



2629/1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants : Tsafrir Ben-Ari :
 Serial No. : 10/596,006 : Group Art Unit #
 Filed : 19 April 2007
 Publication No.: 2008/0048931 :
 Publication Date : 28 February 2008 : Examiner: Unknown
 Based Upon : PCT/IL2004/001067 :
 PCT Filing Date: 18 November 2004 :
 Title : HELMET SYSTEM FOR INFORMATION
 FOR WEAPON SYSTEMS

OK to file 99
Enter
4/18/08

THIRD-PARTY SUBMISSION IN PUBLISHED APPLICATION
UNDER 37 C.F.R. SECTION 1.99

Mail Stop FEE
 Honorable Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

Sir:

The undersigned attorney hereby submits a Third-Party Submission in a Published Application relevant to the above-referenced pending published application and request that such be entered into the application file.

The above-referenced patent application was published on 28 February 2008 and this submission is being filed within two months from the date of publication of the application.

03/28/2008 AYONDAD1 00000073 10596006

01 FC:1806

180.00 OP

The following list of Patents and Publications submitted for the consideration by the USPTO are as follows:

LIST

- (1) Publication entitled: "Tracking: Beyond 15 Minutes of Thought - SIGGRAPH 2001 Course 11", Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, (August 12-17, 2001), pp. 46-55;
- (2) U.S. Patent No. 6,474,159;
- (3) U.S. Patent No. 5,819,206;
- (4) Publication entitled: "ViewPoint EyeTracker Users Guide", Author: Arrington, Karl Frederick, May 25, 1997; and
- (5) Publication entitled: "SR Research EyeLink II", Product Brochure by SR Research Ltd., Circa 2002.

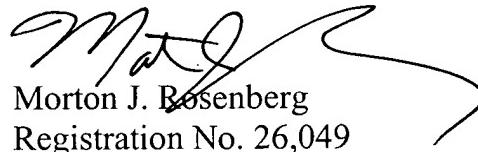
A copy in English of each listed patent application and publication is attached to this submission.

A check in the amount of \$180.00 is attached in payment of the fee set forth in Section 1.17(p).

This submission is being served on the attorneys for Applicant, namely: Dr. Mark M. Friedman, c/o Bill Polkinghorn - Discovery Dispatch, 9003 Florin Way, Upper Marlboro, Maryland 20772, who are believed to be the attorneys for the Applicant and, Rafael-Armament Development Authority Ltd., having an address at: P.O.B. 2250, Haifa, Israel, who is believed to be the Assignee for this application.

In the event there are any further charges associated with the filing of this submission under 37 CFR 1.99, the Honorable Commissioner of Patents is hereby authorized to charge Deposit Account 18-2011 for such charges.

Respectfully submitted,
FOR: ROSENBERG, KLEIN & LEE


Morton J. Rosenberg
Registration No. 26,049

Dated: 26 March 2008

Suite 101
3458 Ellicott Center Drive
Ellicott City, MD 21043
Tel: 410-465-6678

CERTIFICATE OF SERVICE

I hereby certify that on the 26 day of March, 2008
a true copy of the foregoing was mailed via First Class Mail to the following:

Dr. Mark M. Friedman
c/o Bill Polkinghorn - Discovery
Dispatch
9003 Florin Way
Upper Marlboro, MD 20772

Rafael-Armament Development
Authority Ltd.
P.O.B. 2250
Haifa, Israel


Morton J. Rosenberg

SIGGRAPH 2001

Course 11

Tracking: Beyond 15 Minutes of Thought

B. Danette Allen²

Gary Bishop^{1,2}

Greg Welch^{1,2}

**University of North Carolina at Chapel Hill
Department of Computer Science
Chapel Hill, NC 27599-3175**

**<http://www.cs.unc.edu/~{bdallen, gb, welch}>
{bdallen, gb, welch}@cs.unc.edu**

**©Copyright 2001 by ACM, Inc.
<http://info.acm.org/pubs/toc/CRnotice.html>**

**1. Organizer
2. Presenter**

TABLE OF CONTENTS

TABLE OF CONTENTS	3
LIST OF ABBREVIATIONS.....	5
Preface.....	7
Course Syllabus	8
1. Introduction	8
1.1 Course Description	9
1.2 Speaker/Author Biographies.....	9
1.3 Acknowledgements.....	10
2. Background	11
2.1 Basic Coordinate Transforms	11
2.2 Probability and Random Variables.....	21
3. Classifications of Devices and Systems.....	29
3.1 By Physical Medium.....	29
3.2 Sensor Configurations.....	50
3.3 Hybrid Systems.....	54
4. Approaches.....	57
4.1 Traditional Closed-Form Approaches.....	57
4.2 Stochastic Approaches	61
5. Problems and Insights	69
5.1 Classification of Error.....	69
5.2 Total Tracker Error.....	73
5.3 Motion Prediction	75
A. An Introduction to the Kalman Filter	79
A.1 The Discrete Kalman Filter	79
A.2 The Extended Kalman Filter (EKF).....	84
A.3 An Example: Estimating a Random Constant	89
B. Tracking Bibliography	95
C. Related Papers	115

LIST OF ABBREVIATIONS

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
6D	six-dimensional
A/D	analog-to-digital
AR	Augmented Reality
cm	centimeter
CS	coordinate system
DLP	digital light projector
DOF	degree of freedom
EKF	extended Kalman filter
ft	feet
GPS	Global Positioning System
Hz	Hertz
HMD	head-mounted display
kHz	kilohertz
KF	Kalman filter
LCD	liquid crystal display
LED	light emitting diode
m	meter
mm	millimeter
ms	millisecond
MCAAT	multiple-constraints-at-a-time
PV	position-velocity
RMS	root-mean-square
s	second
SCAAT	single-constraint-at-a-time
UNC-CH	University of North Carolina at Chapel Hill
VE	virtual environment
VR	Virtual Reality

Preface

Nearly everyone of a technical bent who has thought about the problem of tracking for graphics for 15 minutes or so believes they have an easy answer—"Why don't you just..." This course (new for SIGGRAPH 2001) is designed to take you beyond those first few minutes of thought with an "under the hood" look at how these systems work. Our goal is to convey a basic understanding of the characteristics of the available technologies, their fundamental limitations, in what cases those limitations hurt you, and some things you can do to improve your results.

In putting together this course pack we decided not to simply include copies of the slides for the course presentation, but to attempt to put together a small booklet of information that could stand by itself. The course slides and other useful information, including an electronic bibliographic version of "Tracking Bibliography" on page 95 are available at

<http://www.cs.unc.edu/~tracker/ref/s2001/tracker/>

We expect that you (the reader) have a basic mathematical background, sufficient to understand explanations involving beginning statistics, random signals, and geometric transformations.

Course Syllabus

Time	Speaker	Topic	Time
1:30 PM	Bishop	Welcome and Introduction	0:15
1:45 PM	Allen	Tracking technologies	0:25
2:10 PM	Bishop	Source/sensor configurations	0:20
2:30 PM	Welch	User and sensor uncertainty/information	0:30
3:00 PM	-	Break	0:15
3:15 PM	Bishop	Traditional approaches	0:20
3:35 PM	Welch	Stochastic approaches	0:25
4:00 PM	Welch	Error sources (spatial and temporal)	0:25
4:25 PM	Bishop	Motion prediction	0:20
4:45 PM	Bishop	Conclusions (summary, resources, etc.)	0:15
5:00 PM			
		Total time	3:30

1. Introduction

One of the important problems in Virtual Environment (VE) research today is that of providing a fast, accurate, and unobtrusive method for reliably *tracking* a computer user's real-world position and orientation or *pose*. Such tracking is necessary in VE systems because a user must continually be provided with two-dimensional computer generated images that match the user's three-dimensional real-world position and orientation. Similarly, human *motion capture* systems are often used for physiological studies (e.g., analysis of athletic motion) or special effects in motion pictures. Usually if the user's position and orientation are not tracked accurately or fast enough, disturbing or even harmful effects can be observed.

Systems for tracking and motion capture for interactive computer graphics have been explored for over 30 years [Sutherland, 1968 #297]. Throughout the years commercial and research teams have explored mechanical, magnetic, acoustic, inertial, and optical technologies. Complete historical surveys include [Bhatnagar, 1993 #264; Burdea, 1994 #267; Mulder, 1994 #286; Mulder, 1998 #288; Meyer, 1991 #432; Meyer, 1992 #285]. Commercial magnetic tracking systems for example [Ascension, 2000 #251; Polhemus, 2000 #294] have enjoyed popularity as a result of a small user-worn component and relative ease of use. Optical systems have been developed for 3D *motion capture* [Woltring, 1974 #310; Woltring, 1976 #485; MAC, 2000 #282] and 6D tracking for visual simulations via head-worn displays [Wang, 1990 #301; Wang, 1990 #538; Ward, 1992 #540; Welch, 1999 #307; Welch, 2001 #308]. Recently inertial *hybrid systems* have been gaining popularity because of the reduced high-frequency noise and direct measurements of derivatives [Foxlin, 1998 #270; Intersense, 2000 #277].

1.1 Course Description

Every year, dozens of vendors display different systems for motion capture and tracking at the SIGGRAPH exhibition, while researchers continue to pursue new approaches in the laboratory. Why are there so many different approaches to this seemingly simple problem? How do the systems differ? What are the strong and weak points of each? How can you decide which is appropriate for your application?

We will attempt to answer these questions and more in this course on some fundamental technologies behind tracking and motion-capture systems. We will use actual systems (commercially available) as examples, describing some algorithms used in popular magnetic, inertial, and optical tracking systems, relating the pros and cons of the systems to the fundamental technologies and the algorithms. While there have been previous SIGGRAPH courses on motion capture, they have primarily concentrated on the graphics

application. Instead we will take you “under the hood” of several systems so that you might better understand the performance (or lack thereof) that you experience with different applications, and perhaps improve your results by adjusting your setup to better match the technology.

1.2 Speaker/Author Biographies

Danette Allen is a Ph.D. Candidate in the Department of Computer Science at the University of North Carolina at Chapel Hill. Her primary research interest is tracking technologies but her interests also include hardware and software for man-machine interaction and virtual environments. Allen graduated from North Carolina State University with degrees in Electrical Engineering and Computer Engineering in 1988 and 1989. She received her Master’s Diploma in Business Administration from Manchester Business School, U.K. in 1990. In 1997, she received an M.E. in computer engineering from Old Dominion University. Allen began working at NASA Langley Research Center (LaRC) in Hampton, Virginia in 1991 where she is currently employed. She is a recipient of NASA’s Silver Snoopy award, the astronauts’ award for outstanding performance in flight safety and mission success. She is a member of the IEEE Computer Society and the Association of Computing Machinery.

Gary Bishop is an Associate Professor in the Department of Computer Science at the University of North Carolina at Chapel Hill. His research interests include hardware and software for man-machine interaction, 3D interactive computer graphics, virtual environments, tracking technologies, and image-based rendering. Bishop graduated with highest honors from the Southern Technical Institute in Marietta, Georgia, with a degree in Electrical Engineering Technology in 1976. He completed his Ph.D. in computer science at UNC-Chapel Hill in 1984. Afterwards he worked for Bell Laboratories and Sun Microsystems before returning to UNC in 1991.

Greg Welch is a Research Assistant Professor in the Department of Computer Science at the University of North Carolina at Chapel Hill. His research interests include hardware and software for man-machine interaction, 3D interactive computer graphics, virtual environments, tracking technologies, tele-immersion, and projector-based graphics. Welch graduated with *highest distinction* from Purdue University with a degree in Electrical Engineering Technology in 1986 and received a Ph.D. in computer science from UNC-Chapel Hill in 1996. Before coming to UNC he worked at NASA’s Jet Propulsion Laboratory and Northrop-Grumman’s Defense Systems Division. He is a member of the IEEE Computer Society and the Association of Computing Machinery.

1.3 Acknowledgements

We thank Leandra Vicci (UNC-Chapel Hill), Richard Holloway (3rdTech, Inc.), and Warren Robinett for their valuable contributions to this course pack. In particular we thank Leandra for her contributions to Section 3.1 on classifications of tracking approaches by physical medium, and Rich for permission to make use of material from his Ph.D. dissertation [Holloway, 1995 #161] in Section 5.1 on tracking error.

2. Background

2.1 Basic Coordinate Transforms

This section briefly discusses the basic 2D and 3D geometrical transformations used in computer graphics and tracking. Most of the information is taken from [Foley, 1997 #562] and [Robinett, 1994 #565], the latter of which is included in Appendix C of this course pack.

We assume a right-handed coordinate system as shown in Figure 2.1.

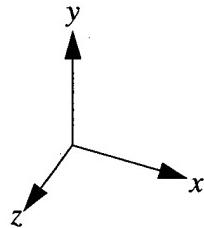


Figure 2.1: Right-handed coordinate system

Translation, rotation and scaling are the essential transformations in computer graphics and tracking. Translation displaces points by a fixed distance in a given direction. Scaling increases or decreases the size of an object. Rotation revolves a point around a specified axis. Figure 2.2 illustrates these transformations in 2D space.

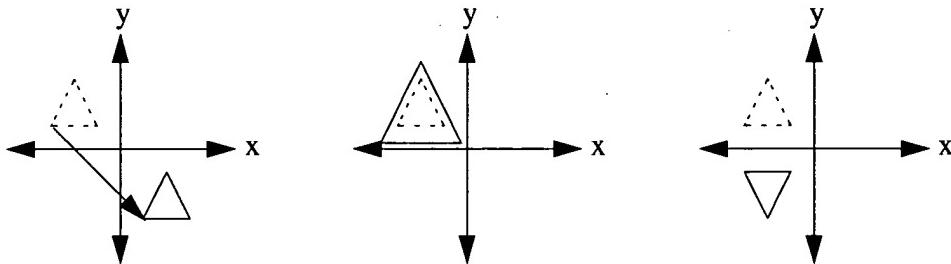


Figure 2.2: Translation, scaling and rotation (around the x-axis) in 2D

In a right-handed coordinate system, positive rotations are defined such that a 90° counterclockwise rotation transforms one positive axis into another. Table 2.1 [Foley, 1997 #562] follows from this convention.

Axis of Rotation	Direction of Positive Rotation
x	y to z
y	z to x
z	x to y

Table 2.1: Positive Rotations

2.1.1 Coordinate Systems

Coordinate systems serve as a 6D basis to which all points, lines, objects, etc. are referenced. For example, in describing the human head, we might declare a head coordinate system with its origin at the extreme tip of the nose. From this origin, we can determine through a series of transformations (translation, rotation and scaling) where the eyes, ears, mouth, etc. are located. Beyond this we can think about a world coordinate system which references the origin of the head coordinate system in the world.

Tracker measurements are provided in the *tracker coordinate system*. Consider an acoustic tracker that provides a range measurement d [m] from room-mounted transmitters to a sensor mounted on the user. This d , measured from the transmitter to the sensor, is referenced to the tracker's coordinate system. What we *want* is the user's head position in room or world coordinates. This is further complicated if we want to know the position of the user's eyes. We must transform from the tracker's coordinate frame to the coordinate frame of the user's head. If we are rendering in stereo, from the user's head coordinate frame, we transform to the both of the user's eyes.

2.1.2 Affine Transformations

With respect to computer graphics, we use the term *transform* to refer to the mathematical operation of modifying a graphics primitive by adding, multiplying, and even dividing its numerical elements achieve such effects as translation, rotation, and perspective projection. In particular, translation, rotation and scaling are classified as *affine* transformations. They preserve parallelism of lines but not angles and lengths. Another less often used affine transformation is a form of distortion known as *shearing*.

2.1.2.1 2D Transformations

We define a point in 2D space as a column vector,

$$\vec{P} = \begin{bmatrix} x \\ y \end{bmatrix}$$

and a transformed point as

$$\vec{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}.$$

We left-multiply a point vector, \vec{P} , by a transformation matrix, M , to acquire a new point vector, \vec{P}' , such that $\vec{P}' = M\vec{P}$.

We can translate points to new positions by adding distances, d , to the coordinates of the original points. We define a translation vector, T , such that

$$T = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

and the transformed point $\vec{P}' = \vec{P} + T$. where

$$x' = x + d_x$$

and

$$y' = y + d_y.$$

Points can be scaled (stretched or shrunk) in the x and y directions by a scaling matrix, S , such that

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

and

$$x' = x \cdot s_x$$

and

$$y' = y \cdot s_y.$$

Points can be rotated around the origin by an angle, θ , with a rotation matrix, R , where

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

and

$$x' = x \cdot \cos\theta - y \cdot \sin\theta$$

and

$$y' = x \cdot \sin\theta + y \cdot \cos\theta.$$

Any and all of the transformations can be applied multiple times and in succession. For example, we can translate a point, scale it, rotate it and translate it again with

$$\vec{P}' = T2 + (R \cdot S \cdot (T1 + \vec{P})).$$

Suppose we have the following object as shown in Figure 2.3 [Foley, 1997 #562]

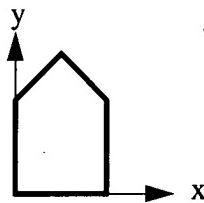


Figure 2.3: Object to be transformed

The series of transformations described above might appear as shown in Figure 2.4.

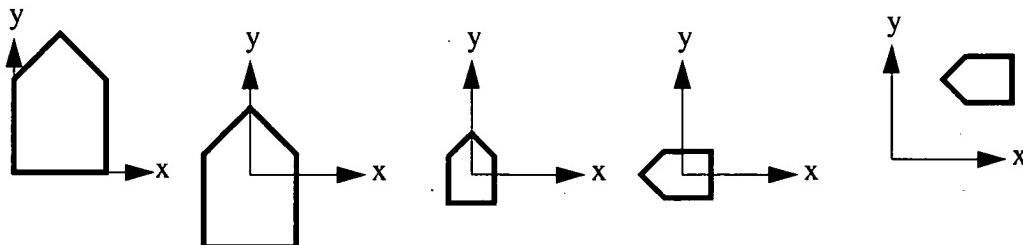


Figure 2.4: Series of transformations on an object

2.1.2.2 Homogeneous Coordinates

Notice that rotation, scaling and shearing are all multiplicative transforms while translation is an additive transform. We would like to be able to treat these transformations consistently to simplify transformation combinations. Homogeneous coordinates allow us

to apply all four transformations multiplicatively. In homogeneous coordinates, we add a third coordinate in 2D space, w . Now a single point, \vec{P} , is represented as a three-element column vector where

$$\vec{P} = \begin{bmatrix} x \\ y \\ w \end{bmatrix} \text{ where } w \neq 0 \text{ and typically } w = 1.$$

When we add an “extra” dimension to the 2D coordinates, every 2D point $[x, y]^T$ in 3D space represents a point along a line that passes through $[x, y, w]^T$ where we want the specific point $[x_1, y_1, w_1]^T$ where $w_1 = 1$. If w does not equal 1, we simply divide all three elements by w to force $w = 1$.

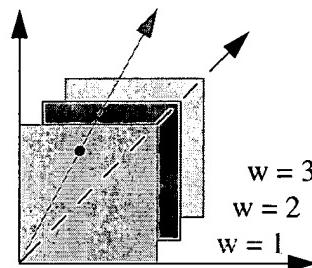


Figure 2.5: 2D homogeneous coordinate space

Any two sets of points $[x_i, y_i, w_i]^T$ and $[x_j, y_j, w_j]^T$ represent the same point if one is a multiple of the other. Dividing P_i and P_j by w_i and w_j respectively will result in the same coordinates $[x, y, 1]^T$. Points at $w = 0$ are points at infinity.

In 2D homogeneous coordinates, we have the following transformation matrices for translation, scaling and rotation:

$$T = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now we can translate a point, scale it, rotate it and translate it again with

$$\vec{P}' = T_2 \cdot R \cdot S \cdot T_1 \cdot \vec{P}.$$

2.1.2.3 3D Transformations

We extend the idea of homogeneous coordinates to 3D space and represent a point as a 4-element vector, \vec{P} , such that

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \text{ where } w \neq 0 \text{ and typically } w = 1.$$

and we have the following transformation matrices for translation and scaling:

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation around a three-coordinate axis is described in the following section.

2.1.3 Representing and working with orientation and rotations

We represent orientation using a rotation from some known orientation just like we represent positions as a translation from some known position origin. The important difference between orientation and position is that orientation space is wrapped on itself in a way that linear position space is not. For example a rotation about the X axis of 45 degrees will produce the same orientation as a rotation of 405 degrees about the X axis. To make matters more confusing there are also combinations of rotations about Y and Z that can produce the same final orientation.

This wrapped nature of orientations is a constant source of difficulty when we attempt to implement even simple operations such as linear interpolation and filtering. We must be prepared to deal with apparent discontinuities in orientation that are really not discontinuities at all but rather differing ways to get to the same place.

Rotation Matrix

The most commonly used representation for rotations is the rotation matrix as described in the previous section on 3D transforms. The upper 3 by 3 sub matrix of the 4 by 4 homogenous rotation transform is a 3D rotation matrix. Its inverse is equal to its transpose and its determinant is 1. We can usefully interpret the columns of a rotation matrix as the new coordinate axes projected onto the old (or origin) coordinate axes.

The rotation matrix is the representation of choice for transforming points because only a simple and efficiently implemented matrix multiply is required. On the other hand, they are inappropriate for filtering or interpolation. Addition and subtraction of elements will almost certainly result in a matrix that is not a proper rotation.

Euler Angles

General rotations in 3D can be expressed as three successive rotations about different axes. For example, a transformation from reference axes to a new coordinate frame may be expressed as follows:

$$\begin{aligned} \text{rotation } \psi \text{ about } z \text{ axis, } R_1 &= \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \text{rotation } \theta \text{ about } y \text{ axis, } R_2 &= \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \\ \text{rotation } \phi \text{ about } x \text{ axis, } R_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \end{aligned} \quad (2.1)$$

Finally, the full transformation can be expressed as the product of these three separate transformations.

$$R = R_3 R_2 R_1$$

$$R = \begin{bmatrix} \cos\psi \cos\theta & \sin\psi \cos\theta & -\sin\theta \\ \sin\phi \cos\psi \sin\theta - \cos\phi \sin\psi & \cos\phi \cos\psi + \sin\phi \sin\psi \sin\theta & \sin\phi \cos\theta \\ \cos\phi \cos\psi \sin\theta + \sin\phi \sin\psi & \cos\phi \sin\psi \sin\theta - \sin\phi \cos\psi & \cos\phi \cos\theta \end{bmatrix} \quad (2.2)$$

Notice the asymmetry in the above matrix in relationship to the three angles. The order of the composition matters because matrix multiplication is not commutative. The mathematics is trying to tell us that the axes interact. Unfortunately the three Euler angles don't indicate this to us at all. If we attempt to filter or interpolate the three angles independently we are ignoring exactly this critical interaction.

Yaw, Pitch, and Roll

People often used the words *yaw*, *pitch*, and *roll* to refer to orientations about a self-referenced coordinate frame. (Historically these terms have been used in navigation such as on ships and in planes.) Specifically if you are sitting upright, looking straight ahead, *yaw* would refer to rotating your head to the left or right around the axis of your neck and spine, *pitch* would refer to elevating or declining your chin up or down, and *roll* would refer to leaning your head toward one shoulder or the other. In other words if you place a right-handed coordinate system at the base of your head such that the Z axis is up and you are looking down the Y axis, yaw, pitch, and roll would correspond to rotation about the Z, X, and Y axes respectively.

Quaternions

Hamilton [Hamilton, 1853 #383] invented quaternions to enable division of vectors. You can find excellent introductions to quaternions in a SIGGRAPH paper by Shoemake [Shoemake, 1985 #551] and in the book “Quaternions and Rotation Sequences” by Kuipers [Kuipers, 1998 #553]. We have also included [Vicci, 2001 #566] in Appendix C.

A quaternion Q consists of a vector augmented by a real number to make a four-element entity. It has a real part Q_r and a vector part Q_v . If Q_r is zero, Q represents an ordinary vector; if Q_v is zero, it represents an ordinary real number. A unit quaternion has the sum of the squares of its four elements equal to 1.

Unit quaternions represent rotations. The vector part of the quaternion specifies the axis of rotation. The real part of the quaternion is the cosine of half the rotation angle. Thus, a quaternion $\{Q_r, Q_v\}$ represents a rotation of $2 \arccos Q_r$ about the axis Q_v following the right-hand rule.

From the above, we can see that the identity rotation is represented by the quaternion $\{1, [0, 0, 0]\}$ which specifies a rotation of 0 degrees about an unspecified axis. Here are some other simple rotations:

$$90 \text{ degrees about Y} = \left\{ \frac{1}{\sqrt{2}}, \left[0, \frac{1}{\sqrt{2}}, 0 \right] \right\},$$

$$270 \text{ degrees about Z} = \left\{ \frac{-1}{\sqrt{2}}, \left[0, 0, \frac{1}{\sqrt{2}} \right] \right\}.$$

Quaternion addition is accomplished simply by adding like parts; real part to real part and vector part to vector part.

Multiplication of quaternions $P = QR$ is defined as

$$\{P_r, P_v\} = \{Q_r R_r - Q_v \cdot R_v, Q_r R_v + R_t Q_v + Q_v \otimes R_v\} \quad (2.3)$$

This equation says that the real part of the result is the product of the real parts minus the inner product of the vector parts. The vector part of the result is the real part of Q times the vector part of R , plus the real part of R times the vector part of Q , plus the cross product of the vector parts. Quaternion multiplication composes the rotations of the two quaternions.

The inverse of a quaternion has the same real part and the negative of the vector part.

In order to rotate a point S by a quaternion Q we evaluate QSQ^{-1} where the multiplication is quaternion multiplication. In this multiplication, think of the vector S as the vector part of a quaternion with zero real part. Though it isn't obvious, this triple product will always result in a quaternion with a zero real part. The vector part will be the rotated point.

To convert a quaternion $\{w, [x, y, z]\}$ to an equivalent rotation matrix we can evaluate:

$$R = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy + 2wz & 2xz - 2wy \\ 2xy - 2wz & 1 - 2x^2 - 2z^2 & 2yz + 2wx \\ 2xz + 2wy & 2yz - 2wx & 1 - 2x^2 - 2y^2 \end{bmatrix} \quad (2.4)$$

Watch out when you are given four numbers said to represent a quaternion rotation; there is no agreement on the order of the elements. Some systems specify the real part followed by the vector part, others specify the real part last. It is impossible to tell by examining the four elements which is the real part unless the rotation is known.

The unit quaternions can be thought of as points on a 4D sphere. Each point represents a rotation. Each point would represent a unique orientation except for the difficulty that points connected by a line through the center of the sphere (that is Q and $-Q$) represent the same rotation. In a time sequence of quaternions we sometimes see apparent jumps that are actually just a result of this ambiguity. When filtering or interpolating in a sequence it is important to handle these apparent discontinuities.

Interpolations among quaternions are properly carried out with spherical interpolation on the 4-sphere [Shoemake, 1985 #551]. Linear operations are equivalent to moving along a chord of the sphere rather than on the surface. In a small region, the sphere appears to be flat so the locally linear operations are a good approximation. Just be sure to renormalize the quaternions that result. Don't attempt linear operations on quaternions that are far apart in orientation. Vicci [Vicci, 2001 #566] introduces a new approach to averaging rotations and orientations represented as quaternions.

Small Euler Angles

When the angles are very small, the Euler angle rotation matrix above takes on a particularly simple and useful form. For small angles (expressed in radians) $\sin \alpha \rightarrow \alpha$ and $\cos \alpha \rightarrow 1$. The sine approximation has relative error of about 0.5% at 10 degrees and

the error decreases quadratically as the angle gets smaller. The relative error of the cosine approximation is about 3 times that in the sine approximation. The rotation matrix in equation (2.2) reduces with the small angle approximation to

$$R = \begin{bmatrix} 1 & \psi & -\theta \\ \psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix}$$

In this form, the angles behave linearly allowing simple interpolation and filtering. This has become our representation of choice. As described in [Welch, 1997 #304], we represent orientation with two terms, a global orientation represented as a quaternion and a local perturbation of that orientation represented as small Euler angles. All filtering operations are done on the small angles making use of their local linearity. The filtered angles are used to update the global orientation so that the small angle property is preserved.

Small Euler angles also have a simple relationship to quaternions. For small angles the approximate quaternion is:

$$Q = \{1 - \phi^2 - \theta^2 - \psi^2, [\phi, \theta, \psi]\}$$

2.1.4 Coordinate System Transforms

Let M_{A_B} denote a transform (scale, rotation, translation) from coordinate system B to coordinate system A. We represent points as column vectors, therefore $\vec{p}_A = M_{A_B} \cdot \vec{p}_B$ is the transform of the point \vec{p}_B in coordinate system B, by M_{A_B} , to point \vec{p}_A in coordinate system A. The composition of two transforms is given by $M_{A_C} = M_{A_B} \cdot M_{B_C}$ and will transform a point in coordinate system C into coordinate system A. Figure 2.6 [Robinett, 1994 #565] shows a diagram of a point \vec{p} and its coordinates in coordinate systems A and B.

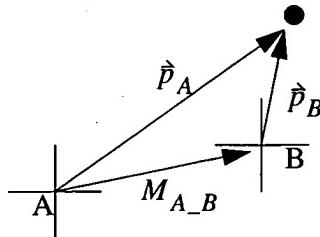


Figure 2.6: Transformation M_{A_B}

We need a sequence of transformations to express the target position in the world coordinate system, a *Head_World* transform. As described in [Robinett, 1994 #565], the primary function of the Head_World transform is to contain the measurement made by the

tracker of head position and orientation, which is updated each display frame as the user's head moves around. The tracker hardware measures the position and orientation of a small movable sensor with respect to the fixed tracker coordinate frame.

The two components of tracker hardware, the tracker's base and the tracker's sensor, have native coordinate systems associated with them by the tracker's hardware and software. If the tracker base is bolted onto the ceiling of the room, this defines a coordinate system for the room with the origin up on the ceiling and with the X, Y, and Z axes pointing whichever way it was mechanically convenient to mount the tracker base onto the ceiling. In a VR environment, the sensor is mounted somewhere on the rigid structure of a head-mounted display, and the HMD inherits the native coordinate system of the sensor.

So, the Head_World Transform is actually comprised of a series of transforms from World to Tracker Base to Head Sensor to Head. The transformation is decomposed into

$$M_{H_W} = M_{H_HS} \cdot M_{HS_TB} \cdot M_{TB_W}$$

We can transform to the user's eyes or any other point in the user's coordinate system with an additional transformation such as M_{LE_H} to transform from the head to the left eye.

2.2 Probability and Random Variables

What follows is a very basic introduction to probability and random variables. For more extensive coverage see for example [Maybeck, 1979 #93; Brown, 1996 #246; Kailath, 2000 #279].

2.2.1 Probability and Random Variables

Most of us have some notion of what is meant by a “random” occurrence, or the probability that some event in a *sample space* will occur. Formally, the probability that the outcome of a discrete event (e.g., a coin flip) will favor a particular event is defined as

$$P(A) = \frac{\text{Possible outcomes favoring event A}}{\text{Total number of possible outcomes}}$$

The probability of an outcome favoring either *A or B* is given by

$$P(A \cup B) = P(A) + P(B). \quad (2.5)$$

If the probability of two outcomes is *independent* (one does not affect the other) then the probability of *both* occurring is the product of their individual probabilities:

$$P(A \cap B) = P(A)P(B). \quad (2.6)$$

For example, if the probability of seeing a “heads” on a coin flip is 1/2, then the probability of seeing “heads” on both of two coins flipped at the same time is 1/4. (Clearly the outcome of one coin flip does not affect the other.)

Finally, the probability of outcome A given an occurrence of outcome B is called the *conditional probability* of A given B , and is defined as

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (2.7)$$

Random Variables

As opposed to discrete events, in the case of tracking and motion capture we are more typically interested with the randomness associated with a *continuous* electrical voltage or perhaps a user's motion. In each case we can think of the item of interest as a *continuous random variable*. A random variable is essentially a function that maps all points in the sample space to real numbers. For example, the continuous random variable $X(t)$ might map time to position. At any point in time t (time is the sample space) $X(t)$ would tell us the expected position.

In the case of continuous random variables, the probability of any *single* discrete event A is in fact 0. That is, $P(A) = 0$. Instead we can only evaluate the probability of events within some interval. A common function representing the probability of random variables is defined as the *cumulative distribution function*:

$$F_X(x) = P(-\infty, x]. \quad (2.8)$$

This function represents the cumulative probability of the continuous random variable X for all (uncountable) events up to and including a . Important properties of the cumulative distribution function are

1. $F_X(x) \rightarrow 0$ as $x \rightarrow -\infty$
2. $F_X(x) \rightarrow 1$ as $x \rightarrow +\infty$
3. $F_X(x)$ is a non-decreasing function of x .

Even more commonly used than equation (2.8) is its derivative, known as the *probability density function*:

$$f_X(x) = \frac{d}{dx}F_X(x). \quad (2.9)$$

Following on the above given properties of the cumulative probability function, the density function also has the following properties:

1. $f_X(x)$ is a non-negative function
2. $\int_{-\infty}^{\infty} f_X(x)dx = 1$.

Finally note that the probability over any interval $[a, b]$ is defined as

$$P_X[a, b] = \int_a^b f_X(x)dx.$$

So rather than summing the probabilities of discrete events as in equation (2.5), for continuous random variables one integrates the probability density function over the interval of interest.

Mean and Variance

Most of us are familiar with the notion of the *average* of a sequence of numbers. For some N samples of a discrete random variable X , the average or *sample mean* is given by

$$\bar{X} = \frac{X_1 + X_2 + \dots + X_N}{N}.$$

Because in tracking we are dealing with continuous signals (with an uncountable sample space) it is useful to think in terms of an *infinite* number of trials, and correspondingly the outcome we would *expect* to see if we sampled the random variable infinitely, each time seeing one of n possible outcomes $x_1 \dots x_n$. In this case, the *expected value* of the discrete random variable could be approximated by averaging probability-weighted events:

$$\bar{X} \approx \frac{(p_1 N)x_1 + (p_2 N)x_2 + \dots + (p_n N)x_n}{N}.$$

In effect, out of N trials, we would expect to see $(p_1 N)$ occurrences of event x_1 , etc. This notion of infinite trials (samples) leads to the conventional definition of *expected value* for *discrete* random variables

$$\text{Expected value of } X = E(X) = \sum_{i=1}^n p_i x_i \quad (2.10)$$

for n possible outcomes $x_1 \dots x_n$ and corresponding probabilities $p_1 \dots p_n$. Similarly for the continuous random variable the expected value is defined as

$$\text{Expected value of } X = E(X) = \int_{-\infty}^{\infty} x f_X(x)dx. \quad (2.11)$$

Finally, we note that equation (2.10) and equation (2.11) can be applied to *functions* of the random variable X as follows:

$$E(g(X)) = \sum_{i=1}^n p_i g(x_i) \quad (2.12)$$

and

$$E(g(X)) = \int_{-\infty}^{\infty} g(x)f_X(x)dx. \quad (2.13)$$

The expected value of a random variable is also known as the *first statistical moment*. We can apply the notion of equation (2.12) or (2.13), letting $g(X) = X^k$, to obtain the k^{th} statistical moment. The k^{th} statistical moment of a continuous random variable X is given by

$$E(X^k) = \int_{-\infty}^{\infty} x^k f_X(x)dx. \quad (2.14)$$

Of particular interest in general, and to us in particular, is the *second moment* of the random variable. The second moment is given by

$$E(X^2) = \int_{-\infty}^{\infty} x^2 f_X(x)dx. \quad (2.15)$$

When we let $g(X) = X - E(X)$ and apply equation (2.15), we get the *variance* of the signal about the mean. In other words,

$$\begin{aligned} \text{Variance } X &= E[(X - E(X))^2] \\ &= E(X^2) - E(X)^2. \end{aligned}$$

Variance is a very useful statistical property for random signals, because if we knew the variance of a signal that was otherwise supposed to be “constant” around some value—the mean, the magnitude of the variance would give us a sense how much jitter or “noise” is in the signal.

The square root of the variance, known as the *standard deviation*, is also a useful statistical unit of measure because while being always positive, it has (as opposed to the variance) the same units as the original signal. The standard deviation is given by

$$\text{Standard deviation of } X = \sigma_X = \sqrt{\text{Variance of } X}.$$

Normal or Gaussian Distribution

A special probability distribution known as the *Normal* or *Gaussian* distribution has historically been popular in modeling random systems for a variety of reasons. As it turns out, many random processes occurring in nature actually appear to be normally distributed, or very close. In fact, under some moderate conditions, it can be proven that a sum of random variables with *any* distribution tends toward a normal distribution. The theorem that formally states this property is called the *central limit theorem* [Maybeck, 1979 #93; Brown, 1996 #246]. Finally, the normal distribution has some nice properties that make it mathematically tractable and even attractive.

The normal distribution is characterized by the following probability density function:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x-m_x)^2\right]$$

where the expected value is

$$m_x = \int_{-\infty}^{\infty} xf_X(x)dx$$

and the squared-variance is

$$\sigma^2 = \int_{-\infty}^{\infty} (x - m_x)^2 f_X(x)dx.$$

Graphically, the normal distribution is what is likely to be familiar as the “bell-shaped” curve shown below in Figure 2.7.

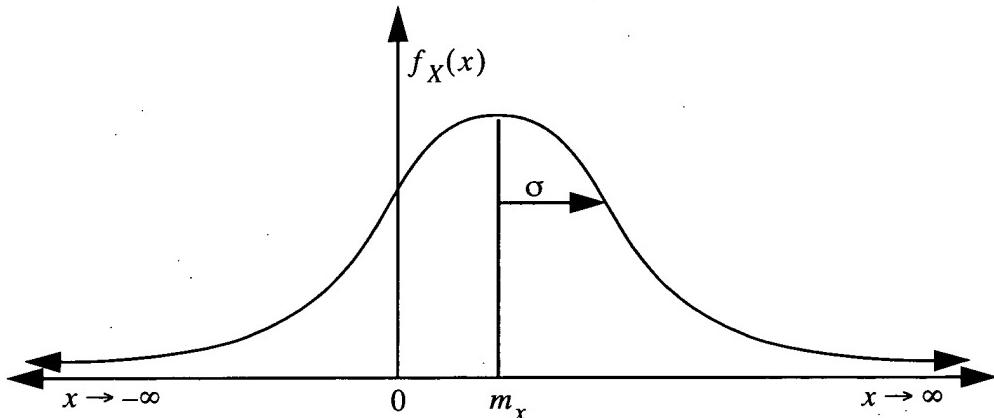


Figure 2.7: The Normal or Gaussian probability distribution function.

Independence and Conditional Probability for Continuous Variables

As with the discrete case and equations (2.6) and (2.7), independence and conditional probability are defined for continuous random variables. Two continuous random variables X and Y are said to be *statistically independent* if their joint probability $f_{XY}(x, y)$ is equal to the product of their individual probabilities. In other words, they are considered independent if

$$f_{XY}(x, y) = f_X(x)f_Y(y).$$

In addition, Bayes' rule follows from (2.7), offering a way to specify the probability density of the random variable X given (in the presence of) random variable Y . Bayes' rule is given as

$$f_{X|Y}(x) = \frac{f_{Y|X}(y)f_X(x)}{f_Y(y)}.$$

Spatial vs. Spectral Signal Characteristics

In the previous section we looked only at the *spatial* characteristics of random signals. As stated earlier, the magnitude of the variance of a signal can give us a sense of how much jitter or "noise" is in the signal. However a signal's variance says nothing about the spacing or the rate of the jitter over *time*. Here we briefly discuss the *temporal* and hence *spectral* characteristics of a random signal. Such discussion can be focused in the time or the frequency domain. We will look briefly at both.

A useful time-related characteristic of a random signal is its *autocorrelation*—its correlation with itself over time. Formally the autocorrelation of a random signal $X(t)$ is defined as

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)] \quad (2.16)$$

for sample times t_1 and t_2 . If the process is *stationary* (the density is invariant with time) then equation (2.16) depends only on the difference $\tau = t_1 - t_2$. In this common case the autocorrelation can be re-written as

$$R_X(\tau) = E[X(t)X(t + \tau)]. \quad (2.17)$$

Two hypothetical autocorrelation functions are shown below in Figure 2.7. Notice how compared to random signal X_2 , random signal X_1 is relatively short and wide. As $|\tau|$ increases (as you move away from $\tau = 0$ at the center of the curve) the autocorrelation signal for X_2 drops off relatively quickly. This indicates that X_2 is less correlated with itself than X_1 .

Clearly the autocorrelation is a function of time, which means that it has a spectral interpretation in the frequency domain also. Again for a stationary process, there is an important temporal-spectral relationship known as the *Wiener-Khinchine relation*:

$$S_X(j\omega) = \Im[R_X(\tau)] = \int_{-\infty}^{\infty} R_X(\tau) e^{-j\omega\tau} d\tau$$

where $\Im[\bullet]$ indicates the Fourier transform, and ω indicates the number of (2π) cycles per second. The function $S_X(j\omega)$ is called the *power spectral density* of the random signal. As you can see, this important relationship ties together the time and frequency spectrum representations of the same signal.

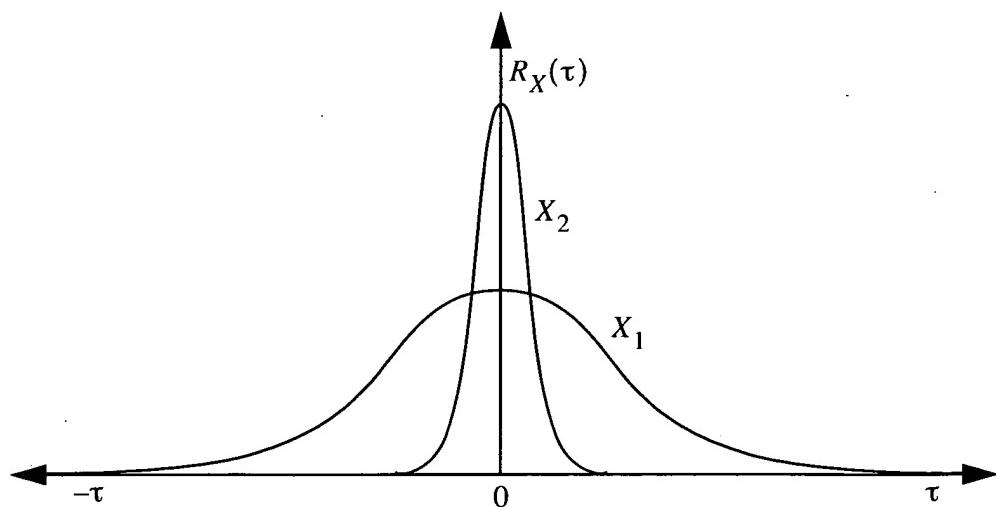


Figure 2.8: Two example (hypothetical) autocorrelation functions X_1 and X_2 .

White Noise

An important case of random signal is the case where the autocorrelation function is a *dirac delta* function $\delta(\tau)$ which has zero value everywhere except when $\tau = 0$. In other words, the case where

$$R_X(\tau) = \begin{cases} \text{if } \tau = 0 \text{ then } A \\ \text{else } 0 \end{cases}$$

for some constant magnitude A . In this special case where the autocorrelation is a “spike” the Fourier transform results in a *constant* frequency spectrum, as shown in Figure 2.9. This is in fact a description of *white noise*, which be thought of both as having power at all

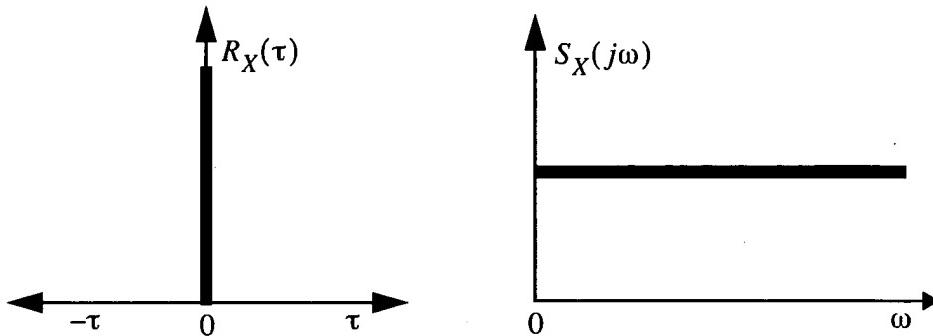


Figure 2.9: White noise shown in both the time (left) and frequency domain (right).

frequencies in the spectrum, and being completely uncorrelated with itself at any time

except the present ($\tau = 0$). This latter interpretation is what leads white noise signals to be called *independent*. Any sample of the signal at one time is completely independent (uncorrelated) from a sample at any other time.

While impossible to achieve or see in practice (no system can exhibit infinite energy throughout an infinite spectrum), white noise is an important building block for design and analysis. Often random signals can be modeled as filtered or *shaped* white noise. Literally this means that one could filter the output of a (hypothetical) white noise source to achieve a non-white or *colored* noise source that is both band-limited in the frequency domain, and more correlated in the time domain.

3. Classifications of Devices and Systems

There are many dimensions to the design space of tracking and motion capture systems, and you can consider the design in the context of any one or more of these dimensions. We like to think about this classification as “many ways to slice the problem.” For example, there is the dimensionality of the information provided by the sources and sensors; the geometric arrangement of the sources and sensors; whether they offer absolute or relative references; passive vs. active; signal to noise ratio; accuracy; resolution; bandwidth; latency; update rate; reliability/repeatability; required infrastructure, and on and on. Here we look primarily at two classifications: by physical medium and by geometric configuration of the devices. We then end this chapter by considering a combination of mediums in *hybrid systems*.

3.1 By Physical Medium

Here we describe the most practical (hence common) physical mediums employed in tracking and motion capture. We thank Leandra Vicci (UNC-Chapel Hill) for her valuable contributions to this section.

3.1.1 Acoustic Tracking

Acoustic trackers typically use ultrasonic sound waves to sense range. Ultrasonic waves have a frequency above the audible range of the human ear of approximately 20,000 [Hz].

3.1.1.1 The Geometry

A single transmitter/receiver pair provides a distance measurement of the target from a fixed point. In the absence of further information, this defines a sphere on whose surface the target is located. As shown in Figure 3.1 [Oceanographers, 2001 #561], the addition of a second receiver or transmitter restricts this surface to the circle of intersection between the two spheres. A third receiver or transmitter restricts this circle to two points, one of which can normally be rejected, and determines a 3D position. Therefore, either three transmitters and one receiver or three receivers and one transmitter are required to find 3D position. To estimate position and orientation, three transmitters and three receivers are required.¹

1. The reader might be familiar with GPS navigation units in which *four* satellites are used for position estimation. The fourth satellite is used to constrain timing differences in the other three.

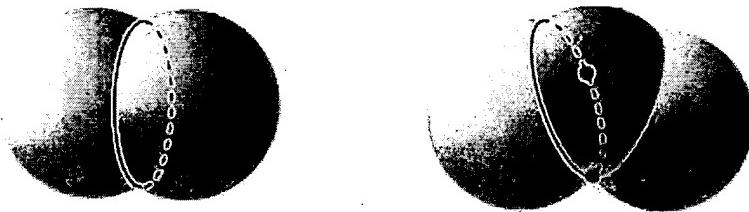


Figure 3.1: Intersection of two spheres (a circle) and three spheres (two points)

3.1.1.2 Techniques

Acoustic trackers typically employ one of two techniques to determine position and orientation:

1. Time of Flight (TOF), and
2. Phase Coherence.

In both methods, the speed of sound is used to convert the time to distance. The drawbacks to any time-of-flight or phase difference method is the inherent delay in waiting for the signal to travel from the source to the destination and this is exaggerated by the slow speed of sound. At 0° C the speed of sounds in air is 331 [m/s], approximately 1.1 [ft/ms]. To complicate matters, the speed of sound varies with temperature and pressure and cannot be treated as a constant. More generally, the speed of sound in a gas is

$$\text{speed} = \sqrt{\frac{\gamma RT}{M}}$$

where γ is a thermodynamic constant of air, R is the ideal gas constant, M is the molecular weight and T is absolute temperature.

Time of Flight (TOF)

The TOF method measures the time t it takes for an ultrasonic pulse to travel from a transmitter to a receiver to provide an absolute distance, d . It takes the time required for a sound wave to travel form the source to the destination and multiplies that by the speed of sound v to get distance.

$$d[m] = v \left[\frac{m}{s} \right] \times t[s]$$

An absolute position is estimated from this distance.

Phase Coherence

The phase coherence method measures the phase difference between the sound wave at the receiver and the transmitter to provide a change in distance, δ . All signals can be represented as a sum of sinusoidal functions of the form $A \cos(\omega t - \phi)$ where ϕ is the phase shift of the signal and A is its amplitude. From this phase difference between the emitted and received signals of a known wavelength or frequency, distance can be computed.

Figure 3.2 below depicts two cosine waves with amplitude of 1 and frequency of 1 Hz. The solid curve is $\cos(x)$ with no phase shift, and the dashed curve is $\cos(x - \pi/4)$ where $\pi/4$ is the phase shift. A phase angle of 360° or 2π [radians] is equal to one cycle.

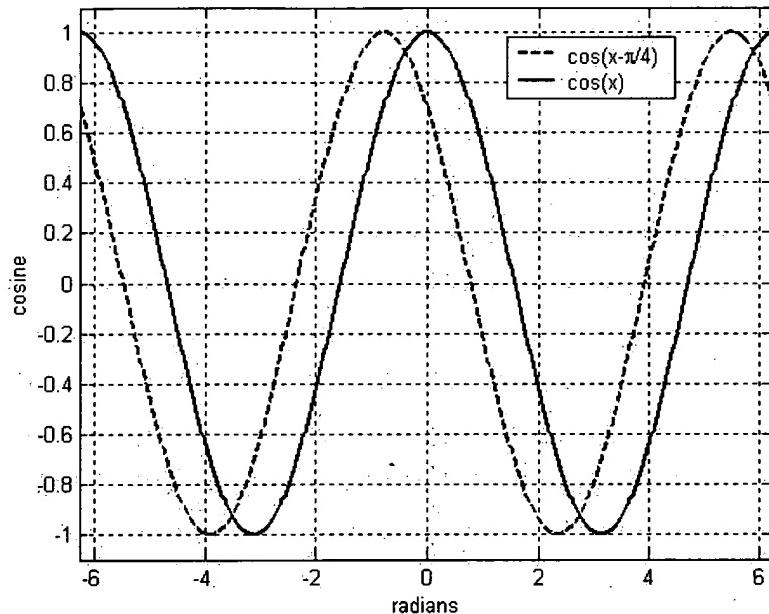


Figure 3.2: Cosine functions with phases of 0 and $\pi/4$ [radians]

The frequency of an acoustic wave, f [Hz], where [Hz] is [1/s], is related to its speed, c [m/s], and wavelength, λ [m], by the equation $c = \lambda f$ where c varies with temperature and pressure as described previously. The fraction δ of the wavelength λ corresponding to the phase shift can be computed by

$$\begin{aligned}\delta[m] &= \lambda[m] \cdot \frac{\phi_{delay}[\text{radians}]}{2\pi[\text{radians}]} \\ &= \frac{c[\text{m/s}]}{f[\text{Hz}]} \cdot \frac{\phi_{delay}[\text{radians}]}{2\pi[\text{radians}]}\end{aligned}$$

If the speed of sound is 331 [m/s] and, as in the graph above, the transmitted signal had zero phase and the received signal had a phase of $\pi/4$ [radians], we calculate a phase difference of $+\pi/4$ [radians] and know that the signal traveled

$$\begin{aligned}\delta[m] &= \frac{c[m/s]}{f[Hz]} \cdot \frac{\phi_{delay}[\text{radians}]}{2\pi[\text{radians}]} \\ &= \frac{331[m/s]}{1[Hz]} \cdot \frac{\frac{\pi}{4}[\text{radians}]}{2\pi[\text{radians}]} \\ &= 331[m] \cdot \frac{1}{8} \\ &= 41.375[m].\end{aligned}$$

If we assume an ultrasonic frequency commonly used in acoustic tracking of 40 [kHz] with the same measured phase shift (1/8 of a cycle), we find

$$\begin{aligned}\delta[m] &= \frac{c[m/s]}{f[Hz]} \cdot \frac{\phi_{delay}[\text{radians}]}{2\pi[\text{radians}]} \\ &= \frac{331[m/s]}{40[kHz]} \cdot \frac{1}{8} \\ &= 1.034[mm].\end{aligned}$$

Notice that a phase of $\phi + (n \cdot 2\pi)$ looks just like a phase of ϕ at the receiver, resulting in an ambiguity in distance. This is usually resolved by assuming phase changes are small between measurement updates. For example, given an acoustic frequency of 40 [kHz], we calculate a wavelength of 8.275 [mm] and measure a phase difference of 0.125 as above. Without previous position information, we cannot determine whether the target is at a distance of 0.125 wavelengths, 17.125 wavelengths or $n + 0.125$ wavelengths. However, if we know the current position estimate is at 100 wavelengths, 0.8275 [m], we assume that the new position is at the closest wavelength count to the current count, 100.125 wavelengths. Therefore the target has moved $\delta = 1.034[mm]$ and the new position estimate of the target is at

$$\begin{aligned}d' &= d + \delta \\ &= 0.8275[m] + 1.034 \times 10^{-3}[m] \\ &= 0.82753[m]\end{aligned}$$

or, simply

$$\begin{aligned}d' &= 100.125[\text{wavelengths}] \cdot 8.275 \left[\frac{\text{mm}}{\text{wavelength}} \right] \\ &= 828.53[\text{mm}] \\ &= 0.82753[\text{m}].\end{aligned}$$

Note that acoustic energy diminishes with the square of the distance between the transmitter and receiver.

3.1.1.3 Commercial and Research Products

Commercial products that employ acoustic sensors include Infusion Systems' FarReach, Intersense's IS-600 Mark 2 and Mark 2 PLUS (inertial hybrid).



Figure 3.3: Intersense's IS-600 Mark 2

Other acoustic trackers include M.I.T.'s Lincoln Wand (1966). See also [Mulder, 1994 #287] for more examples.

3.1.2 Inertial Tracking

Inertial trackers use accelerometers to measure the acceleration for object position and gyros to measure the orientation of object orientation. They are passive, relying on Newton's second law of motion, $F = ma$, and its rotational equivalent $M = I\alpha$, which means there are no physical limits on the working volume and the user is able to move around unencumbered in the environment. Ideally, both are deployed in orthogonal triples (for 3D position in x , y and z and 3D orientation in *roll*, *pitch*, and *yaw*) in order to estimate 6D pose.

3.1.2.1 Accelerometers

Accelerometers actually measure the force exerted on mass since acceleration cannot be measured directly. This measured force, F , for a given mass, m , is transformed into a measure of acceleration, a , by the relationship $F = ma$. The primary transducer of an accelerometer converts acceleration into displacement. Since

$$a = \frac{d^2 r}{dt^2},$$

position r is calculated as

$$r = \iint a dt^2.$$

Accelerometers use a known mass (sometimes called the proof-mass) attached to one end of a damped spring. The other end of the spring is attached to the accelerometer housing. When there is no acceleration imposed upon the accelerometer, the spring is at rest and exhibits zero displacement.

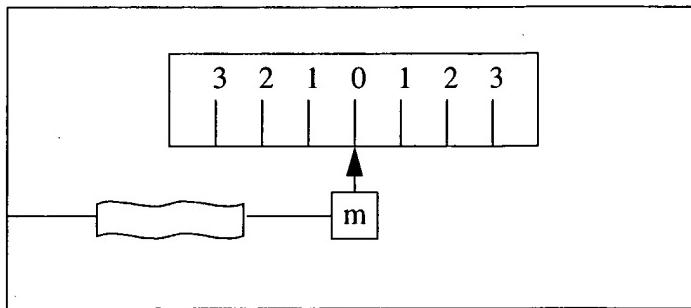


Figure 3.4: Spring at rest with zero displacement.

If a force is applied to the housing, the housing will accelerate but inertia causes the suspended mass to lag behind, resulting in a displacement.

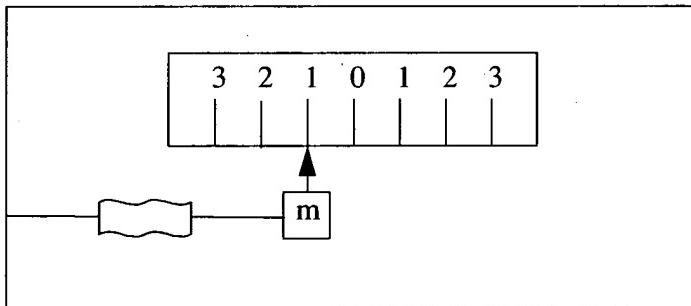


Figure 3.5: Spring under acceleration with displacement.

The displacement of the mass and extension/compression of the spring is proportional to the acceleration of the housing or, in the case of head tracking, the acceleration of the wearer.

A secondary transducer converts this displacement into a usable signal. Two common such transducer types are potentiometric and piezoelectric. Potentiometric devices attach the displacement of the mass to the slider of a potentiometer. The output voltage of a potentiometer is linearly proportional to the slider position. Voltage varies directly with current and the electrical resistance supplied by the potentiometer by $V = IR$ where V is

voltage, I is current and R is resistance. Piezoelectric devices attach the displacement to a piezoelectric element (a piezoelectric crystal that produces an electric charge when a force is exerted upon it) that generates a voltage proportional to the displacement.

3.1.2.2 Gyroscopes

Gyroscopes employ the principle of conservation of angular momentum. If torque is exerted on a spinning mass, its axis of rotation will precess at right angles to both itself and the axis of exerted torque. If the mass spins very fast, it will have a large angular momentum that will strongly resist changes in direction.

Figure 3.6 illustrates the phenomenon of precession. If we fix the axis of rotation on one end of the gyroscope allowing it to pivot around this point, the force of gravity simply causes the gyroscope to fall down in the direction of the gravity vector if it is not spinning. However if the gyroscope is spinning, gravity exerts a torque on the gyroscope about an orthogonal axis to the axis of rotation. If the gyroscope is spinning at a sufficient rate, the gyroscope does not fall in the direction of gravity. Instead it rotates around the axis that is orthogonal to both the gyroscope's axis of rotation and gravity's torque axis.

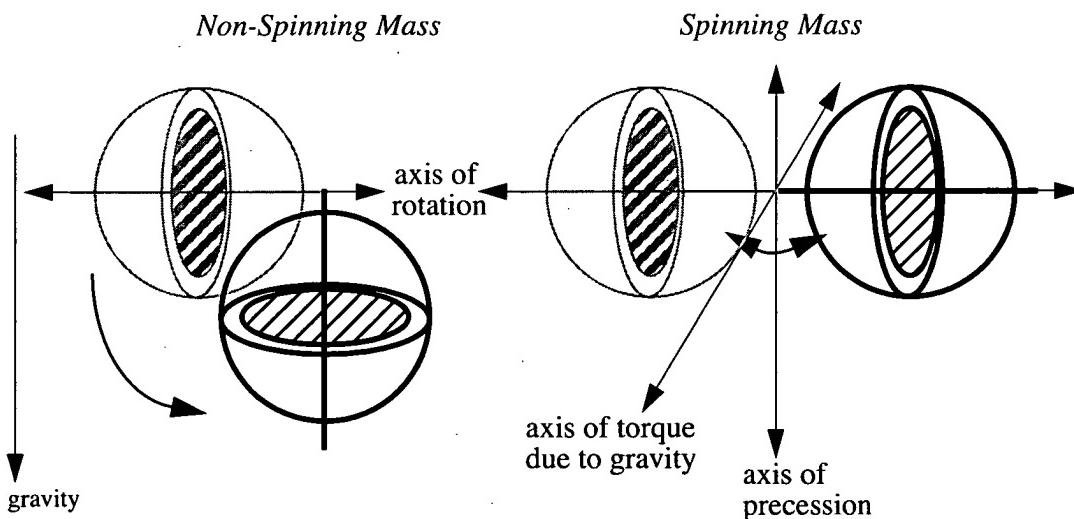


Figure 3.6: Precession

If the spinning mass is mounted on gimbals, these principles can be used to measure changes in direction. The angle that the gyroscope makes with its housing (gimbal deflection) is a measure of the angular momentum or angular velocity. If the gimbals are

constrained with springs, the rate of change of direction can be measured. This configuration is known as a rate gyro. Examples of mechanical rate gyroscopes are shown in Figure 3.7 [Britannica, 1994 #559].

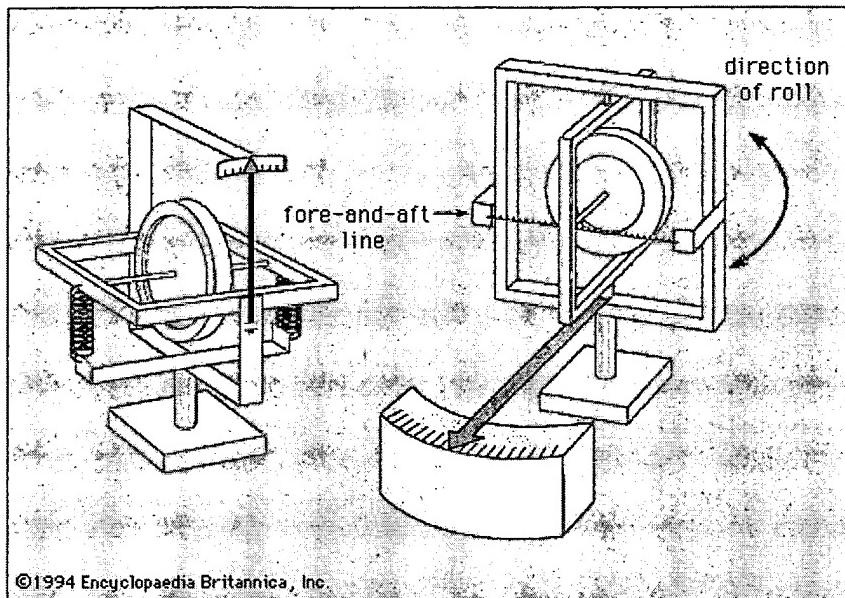


Figure 3.7: Rate Gyroscopes for measuring rate of turn (left) and rate of roll (right)

For 3D orientation (roll, pitch and yaw), three rate gyroscopes are typically fitted to a platform with their axes mutually perpendicular. Two of the gyroscopes provide for horizontal stabilization of the platform--an essential requirement to eliminate the influence of accelerations due to gravity--while the third is responsible for the north-south alignment. Pitch, roll, and yaw are detected by the three gyroscope input axes. The gimbal deflection of each of the gyroscopes is converted into a signal.

The physics supporting inertial measuring devices are most easily illustrated using mechanical devices. While extremely accurate, these devices are not used for human tracking because of their size and mass. Instead micromechanical devices are often employed. An example of is BEI Systron Donner Inertial Division's GyroChip that uses a vibrating piezoelectric quartz tuning fork to sense rate.

Figure 3.8 illustrates the micromechanical gyro designed as an electronically-driven resonator. When an angular rate is applied to its drive tines, Coriolis or torsional forces are exerted on its drive tines which cause its vibrations to couple to the pickup tines. The pickup fork vibrations are detected and used to measure the angular rate.

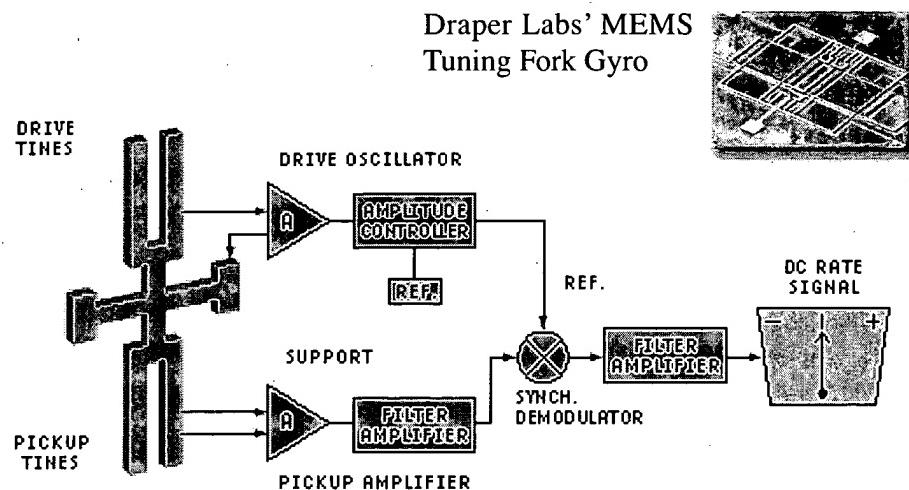


Figure 3.8: BEI Systron Donner Inertial Division's GyroChip technology

Systron Donner [Donner, 2001 #560] offers the following explanation.

The piezoelectric drive tines are driven by an oscillator to vibrate at a precise amplitude, causing the tines to move toward and away from one another at a high frequency. This vibration causes the drive fork to become sensitive to angular rate about an axis parallel to its tines, defining the true input axis of the sensor.

Vibration of the drive tines causes them to act like the arms of a spinning ice skater, where moving them in causes the skater's spin rate to increase, and moving them out causes a decrease in rate. For vibrating tines ("arms"), an applied rotation rate causes a sine wave of torque to be produced, resulting from "Coriolis Acceleration," in turn causing the tines of the Pickup Fork to move up and down (not toward and away from one another) out of the plane of the fork assembly.

The pickup tines thus respond to the oscillating torque by moving in and out of plane, causing electrical output signals to be produced by the Pickup Amplifier. Those signals are amplified and converted into a DC signal proportional to rate by use of a synchronous switch (demodulator) which responds only to the desired rate signals.

The DC output signal of the GyroChip is directly proportional to input rate, reversing sign as the input rate reverses, since the oscillating torque produced by Coriolis reverses phase when the input rate reverses.

Regardless of the technology, both accelerometers and gyros provide derivative measurements. Linear accelerations must be integrated twice and angular rates need to be integrated once to derive position and orientation, respectively. This integration causes these inertial measurements to be sensitive to drift. Position error diverges with time as errors accumulate. This drift can be combated with periodic recalibration by the use of another tracking method that corrects the accumulated error periodically. Conversely inertial tracking performance is very good at high frequency and over short time intervals.

Models useful in explaining the frequency characteristics of an inertial system are shown in Figure 3.9. The dashed box marked "User" contains a question mark to indicate that the user's acceleration (motion) is unknown. The dashed box marked "Accelerometers" shows acceleration measurement noise $\epsilon_a(t)$ being summed with the ideal user acceleration signal, and the sum then being integrated twice to obtain a position estimate. The lower part of the figure shows the corresponding transfer function coefficients for spectral (frequency) analysis.

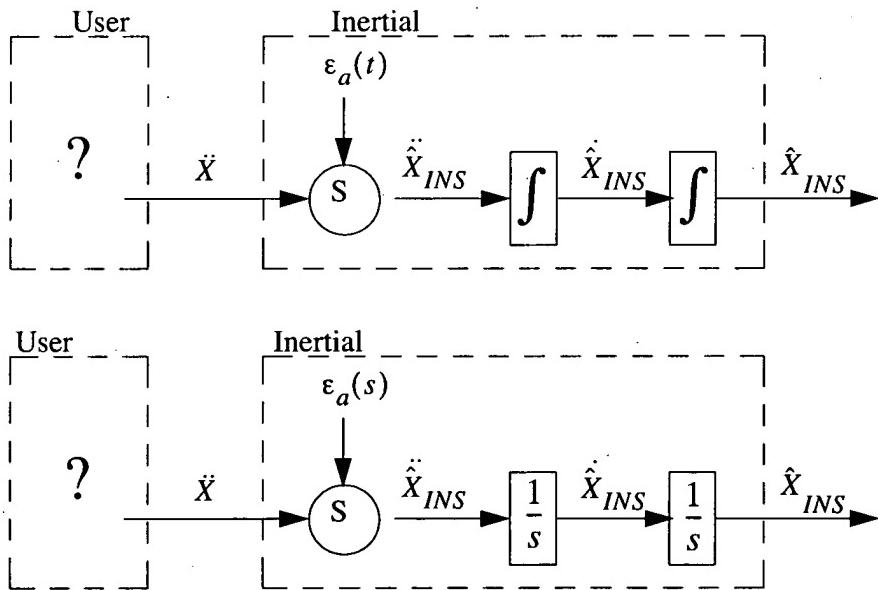


Figure 3.9: Integration in time domain and division in frequency domain of inertial measurements.

The user's true acceleration is their position twice differentiated, i.e. the user's acceleration is weighted by the square of the frequency (s) of their motion. This acceleration signal is then weighted by the inverse square of the frequency (s) as it is integrated twice in the inertial system to obtain a position estimate. The end result is a unity frequency weighting of the position in the final estimate.

From Figure 3.9 we also see that any electrical noise $\varepsilon_{a(t)}$ incurred during the measurement of the accelerometer output is weighted solely by the inverse square of the frequency (s) as it is integrated twice in the inertial system. The end result is an inverse square frequency weighting of electrical measurement noise in the final position estimate.

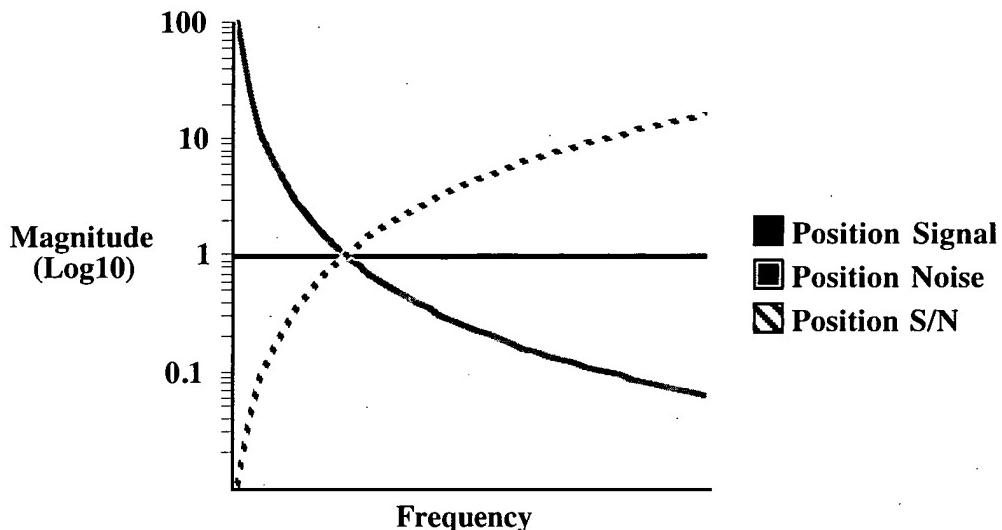


Figure 3.10: Logarithmic plots of typical accelerometer-based position signal vs. noise (for constant velocity).

In Figure 3.10, the estimated position signal, noise, and signal-to-noise ratio for an inertial system using accelerometers are plotted together against frequency. This figure demonstrates why the practical application of a solely inertial-based tracking system is impractical. At low frequencies the position estimate noticeably diverges as measurement noise is erroneously interpreted as acceleration. The most common sources of low frequency noise are the unavoidable and often time-dependent random “DC” (or very low frequency) biases. Such bias errors can cause an inertial-based tracker to report that a subject is moving even when that subject is completely still, or conversely to report that a subject is still when in fact they are slowly moving. The result is that in general for inertial devices to be practical they must be combined with some other mediums as described in Section 3.3.

An additional source of error is misalignment with the gravity vector whose effects must be subtracted out of inertial measurements. The effect of gravity can be significant. One degree of tilt error over ten seconds can cause nine meters of position error.

Noise and quantization error in the signals from the inertial sensors is another important source of error. Figure 3.11 shows how error accumulates in inertial systems. The curves are contours of time to 0.1 [m] of accumulated error for accelerometer (vertical axis) and rate gyro (horizontal axis) signals with the indicated number of useful bits. For example,

with 15 bits of useful dynamic range on the rate gyro signal and about 12 bits of useful range on the accelerometer signal, the graph predicts about 20 seconds of operation before 10 [cm] of error accumulates. Increasing the range of the accelerometers will not improve the system performance because the rate gyros are the limiting factor. The shape of the curves can be explained by realizing that rate gyro errors will result in misestimation of the system tilt. As explained earlier, tilt errors cause fractions of the gravity vector to be integrated as actual acceleration resulting in potentially large position errors. Region 1 includes inertial units readily available today. Region 2 includes today's peak performance. Region 3 reflects arguably unachievable performance.

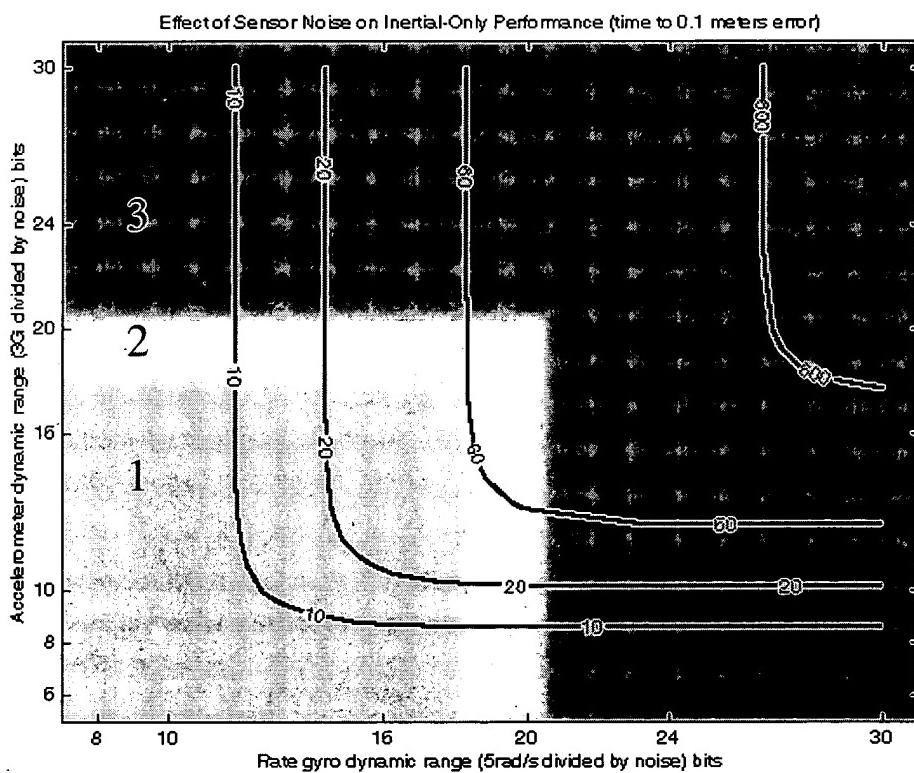
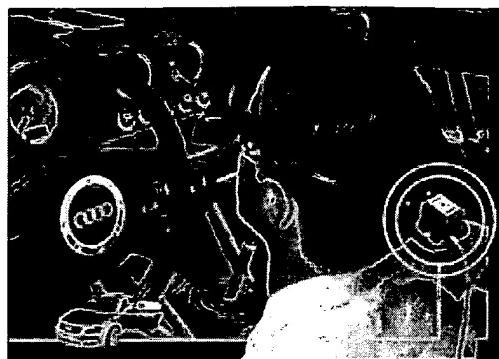


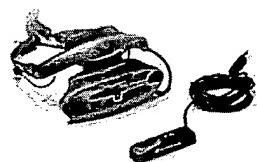
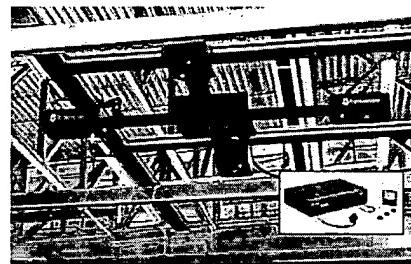
Figure 3.11: Time to 0.1 [m] error

3.1.2.3 Commercial and Research Products

Inertial trackers have become more common over the last few years due to IC technologies allowing for significant reduction in size. Because they require some form of periodic calibration to control drift they are typically used in hybrid tracking products. Commercial products that employ inertial sensors Ascension's 3D-BIRD, Intersense's IS-300/600 and InterTrax 2. See also [Mulder, 1994 #287] for more examples.



Ascension's 3D-BIRD



Intersense's InterTrax



Intersense's IS-300/600

Figure 3.12: Some example inertial tracking systems.

3.1.3 Magnetic Tracking

Magnetic trackers use magnetic fields to measure range and orientation. These magnetic fields can be low frequency AC fields or pulsed DC fields and three orthogonal triaxial coils are used at both the transmitter and receiver to produce position and orientation measurements.

3.1.3.1 Magnetic Fields

Generating magnetic fields

Current carrying coils are used to generate the source magnetic fields. The magnetic field produced by a circular coil of wire carrying a current, I , at a distance d and off-axis angle, θ , is described by

$$H_r = \frac{M}{2\pi d^3} \cos(\theta) \quad (\text{radial component})$$

$$H_\phi = \frac{M}{4\pi d^3} \sin(\theta) \quad (\text{tangential component in } \theta \text{ direction})$$

$$H_\phi = 0 \quad (\text{tangential component in } \phi \text{ direction})$$

where H_r and H_θ are the radial and tangential components of the field, M is the magnetic moment of the loop ($M = NIA$), A and N are the area enclosed by the current loop and number of the turns of the loop or winding [Raab, 1979 #174] and I is the current. Figure 3.13 illustrates the magnetic field of a single-turn winding.

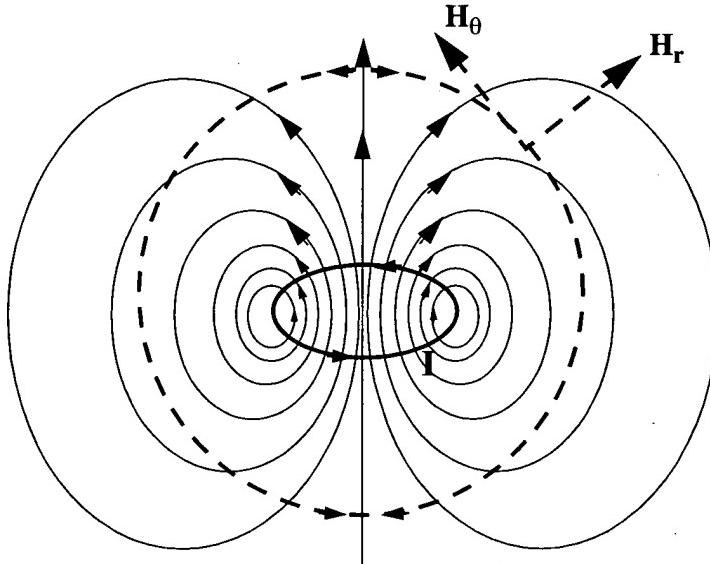


Figure 3.13: Magnetic Dipole

Detecting magnetic fields

A time varying magnetic field will induce a voltage in a coil that can be measured electrically. The magnitude of the voltage is proportional to the area circumscribed by the coil, the rate of change of the field, and varies as $\cos\theta$ where θ is the angle between the direction of the field lines and the axis of the coil.

3.1.3.2 System Configuration

A magnetic tracking system consists of a transmitter and a receiver in the form of coils. A 1D sensor for estimating the position in z (i.e. direction of gravity) is made up of a single coil transmitter oriented in the z-direction. When current is applied to the coil a magnetic field is generated. At the receiver, this induces a maximum voltage proportional to the sensed magnetic field strength in a receiving coil oriented in the same direction as the field.

The induced voltage level provides information about the both distance from the transmitter to the receiver and the axis-alignment between them. Boundaries of equal accuracy are found along a hemisphere or sphere around the transmitter. In Figure 3.14

[Burdea, 1994 #267] the accuracy and amplitude A_2 is less than A_1 where the radius R_2 is greater than R_1 . Here the radius (distance from the transmitter) is the determining factor.

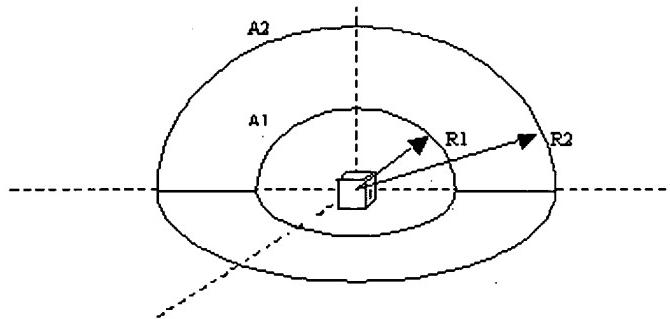


Figure 3.14: Accuracy contours around a coil

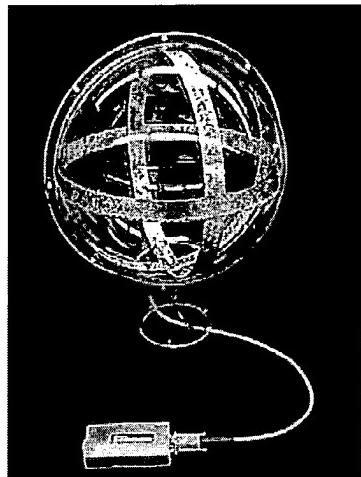
Three separate coils wound orthogonally around a core are used to generate and measure the magnetic field strength in x, y and z. When three orthogonal coils are used, the three source coils are activated serially and the induced signal in each of the three receiving coils is measured. A full measurement cycle contains three measured values for each of the three source coils and this nine-element measurement is used to calculate the position and orientation of the receive coils relative to the source coils. The signal strength per receive coil decreases cubically with distance and with the cosine of the angle between its axis and the local magnetic field direction. The strength of the induced signals can be compared to the known strength of the transmitted signals to find distance. The strength of the induced signals are compared to each other to find orientation.

One disadvantage of AC magnetic sensors is that ferromagnetic and other conducting objects within the sensor space can distort the magnetic field geometry. An eddy current is induced in conducting materials by the source magnetic field (and other fields such as the Earth's magnetic field) and these currents produce small magnetic fields around the conducting materials. The fields cause distortions in the source fields shown above resulting in erroneous pose estimates. This is particularly a problem when using AC transmitters because of the continuously varying nature of AC signals.

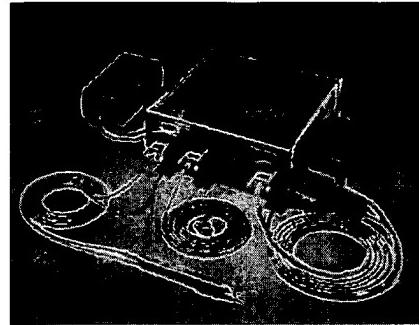
The use of DC transmitters overcomes the eddy current interference problem. Eddy currents are generated only at the beginning of a measurement cycle and a steady state can be reached where the effect of interfering magnetic fields is minimized as the eddy current values approach zero. However, distortions due to ferromagnetism, mainly in steel or iron objects, are still a problem for DC systems.

3.1.3.3 Research and Commercial Products

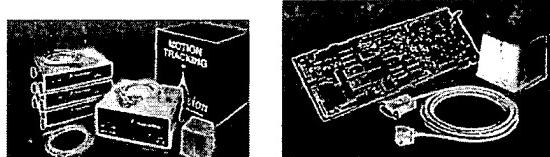
Despite some shortcomings, magnetic tracking systems have historically enjoyed popularity as a result of a small user-worn component and relative ease of use. Commercial products that employ magnetic sensors include Polhemus' 3SPACE FASTRAK and ISOTRAK II (AC electromagnetic) and Ascension's Flock of Birds and PcBIRD (Pulsed-DC electromagnetic). See also [Mulder, 1994 #287] for more examples.



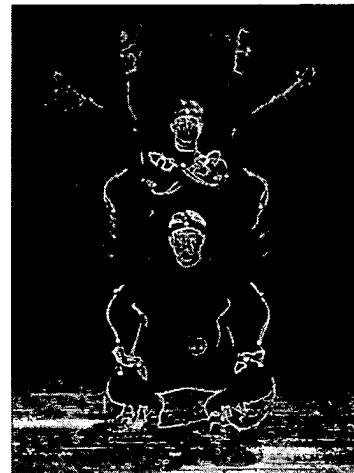
Polhemus' Long Ranger Tracker



Polhemus' Stylus Option



Ascension's Flock of Birds and PcBIRD transmitter



Polhemus' Star*Trak

Figure 3.15: Examples of magnetic tracking systems.

3.1.4 Mechanical Tracking

Mechanical trackers measure joint angles and lengths between joints. Given one known position, all other absolute positions can be derived from the relative joint measurements. They are used to measure all parts of the body and have been historically employed in

motion capture. In implementation, they range from whole body suits that measure the position of all the major joints to mechanical arms to gloves that measure the location of the hands and fingers.

Mechanical tracking systems can be ground-based in which one point of the tracker is affixed to the floor at a known location. This limits the user's range of motion. They can also be body-based in which the system is attached only to the user, typically in the form of an exoskeleton. This does not limit physical range of motion but can be prohibitive if the suit is heavy or bulky.

The rotations and lengths can be measured by gears, potentiometers, and bend sensors as shown in Figure 3.16.

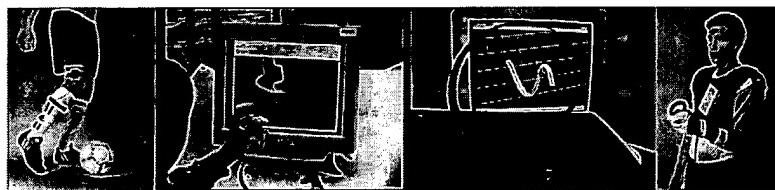


Figure 3.16: Mechanical tracking sensors.

A potentiometer is a device that transduces a rotation or displacement to a voltage. A bend sensor is typically a thin strip of plastic whose resistance changes as it bends. The more it bends, the higher the resistance. Alternatively, optical fiber can be treated for a short distance on one side to lose light proportional to the angle through which it is bent.

An advantage of mechanical trackers is the elegant addition of force feedback as illustrated in Sensable's Phantom and the EXOS dexterous hand master pictured in Figure 3.17.

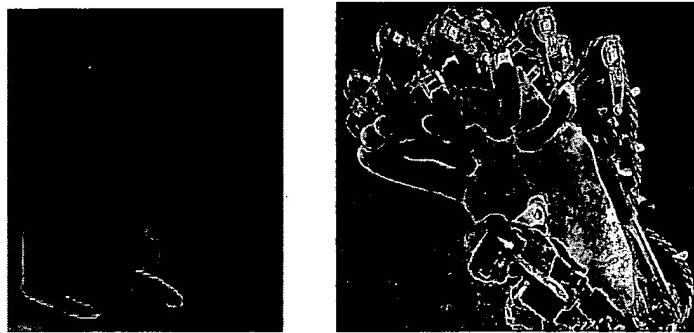
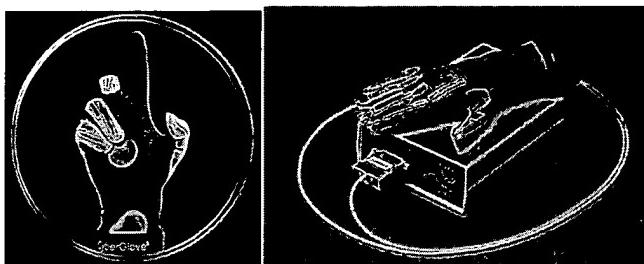


Figure 3.17: Sensable's Phantom and EXOS dexterous hand

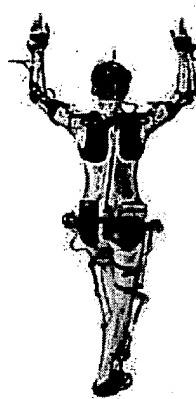
They typically provide good accuracy and low latency but can be cumbersome for the user. The user may be constrained by the suit or arm and, therefore, may not have full freedom of movement.

3.1.4.1 Research and Commercial Products

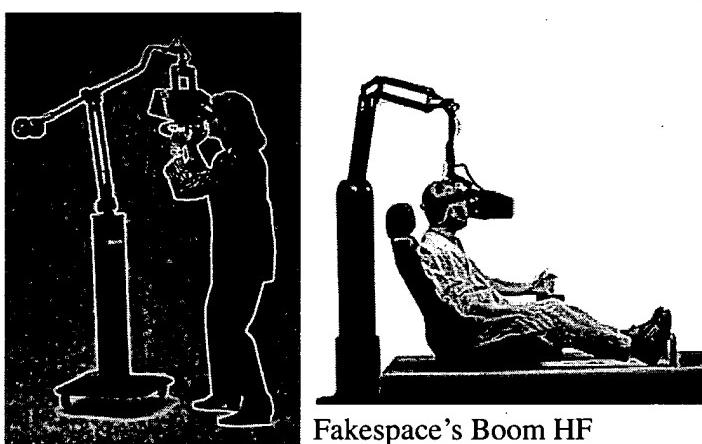
Commercial products that employ mechanical sensors are Fakespace's Boom HF, Virtual Technologies' CyberGlove and MetaMotion's Gypsy (motion capture only). See also [Mulder, 1994 #287] for more examples.



Virtual Technologies' CyberGlove



MetaMotion's Gypsy



Fakespace's Boom HF

Figure 3.18: Some example mechanical tracking systems.

3.1.5 Optical Tracking

Optical trackers use light to measure angles. As shown in Figure 3.19, a single point on the detector, an optical sensor or image plane, provides a ray defined by that pixel and the center of projection, C . As is the case with the acoustic medium, optical energy diminishes with the square of the distance between the transmitter and receiver.

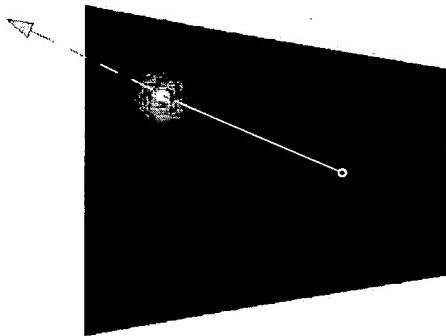


Figure 3.19: Centroid of light.

3.1.6 Targets

Optical tracking systems (also called image-based systems) can use two types of targets: active targets and passive targets. Active targets are powered such as an light-emitting diode (ILED). Infrared LEDs (ILEDs) are used to combat noise due to ambient light. Passive targets are not powered such as reflective materials or high contrast patterns. Regardless of categorization, a detector is used to record the object being tracked (the target) and from this angle measurement, position and orientation can be derived. Some systems use no artificial targets and, instead, use elements of the natural scene.

3.1.7 Detectors

Detectors can be simple video and CCD cameras (typically used with passive targets), or lateral-effect photodiodes (typically used with active targets) that provide the location of the centroid of light on the image plane as illustrated in Figure 3.19. Video cameras and CCDs require imaging techniques to determine position while photodiodes produce currents that are directly proportional to the light center's position.

3.1.7.1 Lateral Effect PhotoDiodes (LEPDs)

1D LEPDs place two terminals on either side of a silicon photosensitive region. An incident light beam produces electrons that flow laterally towards the terminals on either side of the region. The amount of current measured at each terminal is dependent on the distance of the centroid of the incident beam from the terminals. If the centroid occurs at

the center of the region, equal current values will be measured at each terminal. A 2D LEDP sensor is made up of 2 1D sensors rotated 90 degrees from each other. LEPDs can be used to detect both the intensity and position of an incident light beam.

3.1.7.2 Quad Cells

Quad cells are made up of four photosensitive cells. When a light beam is incident upon a quad cell, a current is generated in each of the four quadrants proportional to the amount of light seen by each. A perfectly circular beam illuminating the exact center of the quad cell will produce equal photocurrents in the four quadrants. As shown in Figure 3.20, if the beam is too small such that it falls in between quad cell or too large such that it covers all four cells, there is no information to be gained from the voltage measurements of each cell.

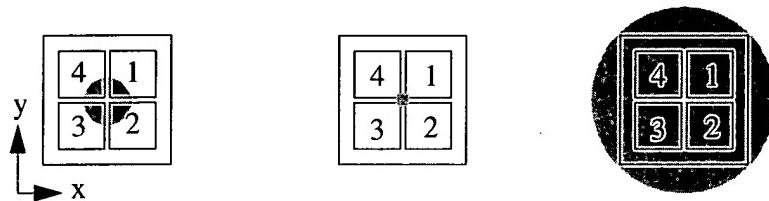


Figure 3.20: A Quad Cell

The x and y displacements of the beam relative to the center of the quad cell can be calculated using the following formulas:

$$x = \frac{(i_1 + i_2) - (i_3 + i_4)}{i_1 + i_2 + i_3 + i_4}$$

$$y = \frac{(i_1 + i_4) - (i_2 + i_3)}{i_1 + i_2 + i_3 + i_4}.$$

3.1.7.3 Charge Coupled Devices (CCDs)

A CCD array can be a 1D or 2D collection of light-sensitive cells.

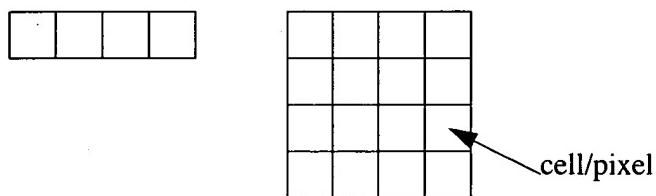


Figure 3.21: 1D and 2D CCD Detector Arrays

When light is incident upon a CCD cell, electrons are produced and each cell accumulates electrons proportional to the amount of incident light for the duration of the dwell time. The electron count per array cell is read out as a voltage to provide a per pixel luminance value. An “empty” cell corresponds to zero volts (black). This collection of pixel values produces a digital image that is similar to the analog images captured with film. The array of luminance values can be analyzed to pinpoint the cell (or pixel) of highest intensity and sub-pixel accuracy can be achieved by interpolation between discrete pixel values. The dwell time dictates how long the CCD will accumulate a charge and must be set large enough so that sufficiently high SNR is achieved for pinpointing the target. However, care should be taken in fixing the dwell time because it affects the update rate of the CCD. Long dwell times delay measurements which affect the rate at which position estimates can be determined.

While optical sensors fundamentally provide angle measurements, they can also be used to determine range. The amount of defocus (i.e. blur) due to the limited depth of field of lenses can provide information about distance. Structures which are closer to the plane of focus for an image will appear sharper. Conversely, structures farther from the plane of focus will appear more blurred. As shown in Figure 3.22 (from [Goshtasby, 2001 #567; Wood, 2000 #568]) there is ambiguity between two points that lie equidistantly on either side of the plane of focus. This ambiguity can be eliminated with the addition of a second image plane.

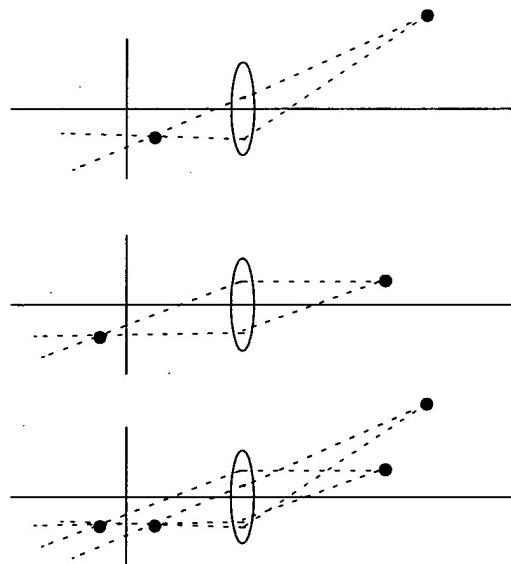


Figure 3.22: Two objects produce the same position and blur on the image

In general, optical tracking systems exhibit high accuracy and resolution and are well suited to real-time systems in terms of update rate (light is a fast medium). However, there is much variety among optical tracking systems due to technical specification difference

(focal length, FOV, etc.), system configuration (outside-in, inside-out) and target type. This accuracy depends on a clear line of sight between the sensor and the target. If something in the environment or the object itself blocks this line-of-sight, optical tracking systems suffer from obscuration difficulties.

In Section 4.1 we describe some traditional approaches to using optical sensors for pose estimation. In Appendix C we have also included a copy of [Welch, 2001 #308], which describes the UNC-Chapel Hill HiBall optical tracking system.

3.1.7.4 Commercial and Research Products

Commercial products that employ optical sensors include InMotion's CODA, 3rdTech's HiBall-3000, University of Iowa's Selspot II, Arcsecond's Vulcan, Ascension's laserBIRD, and Phoenix Technologies' Visualeyez (motion capture only). Other optical trackers include Omniplanar's Virtual Tracker and the Minnesota scanner. See also [Mulder, 1994 #287] for more examples.

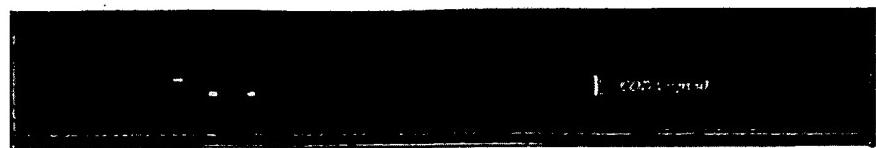
3.2 Sensor Configurations

Choosing the physical medium and sensors for a tracking system determines only a part of its capability. The geometric configuration of the sources and sensors also has a profound effect. For example, in an optical tracker we can have fixed sensors observing moving targets or moving sensors observing fixed targets. This has lead to the descriptions "outside looking in" and "inside looking out" as described in [Welch, 2001 #308], but the direction of "looking" is not the determining factor. Rather, as we shall see, the distinguishing factor is the coordinate frame in which the measurements are made.

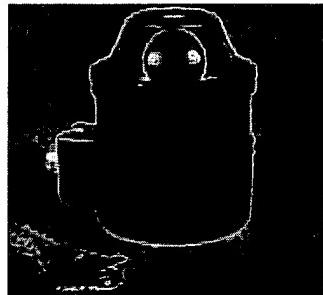
3.2.1 Measurements in the Laboratory Frame

The CODA system [BL, 2000 #266] is a commercial example of a system that makes its measurements in the laboratory coordinate frame as illustrated in Figure 3.24. It uses three or more stationary 1D optical sensors that observe LED beacons that are free to move (a typical "outside looking in" configuration). Each of the 1D sensors narrows the possible location of the LED to a plane in three space. The three planes intersect in a mathematical point which is the system's estimate of the three-dimensional coordinates of the LED. The FlashPoint 5000 [IGT, 2000 #276] from Image Guided Technologies uses the same fundamental measurement strategy though the sensors and optics are quite different. The Selspot and OPTOTRAK [NDI, 2001 #546] systems are also essentially the same though they use at least two 2D sensors (each determining a line) rather than at least three 1D sensors.

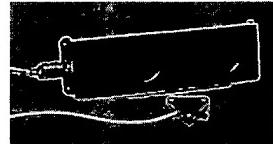
The Minnesota Scanner [Sorensen, 1989 #467] uses spinning mirrors at fixed locations in the laboratory to project planes of light into the working volume. The tracked target is a photo-detector that detects the time at which it is illuminated by the swept plane of light. The time is used along with the precisely known rotation rate of the mirror to determine



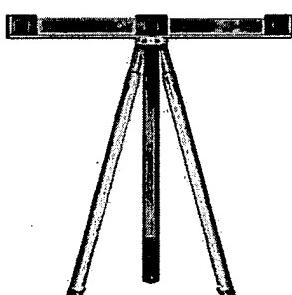
InMotion Systems' CODA



3rd Tech's HiBall-3000



Ascension's LaserBIRD



Phoenix's Visualeyez



Arcsecond's Vulcan

Figure 3.23: Some example optical tracking systems.

the equation of a plane that passes through the detector and the center of rotation of the mirror. Just as in the CODA system, the multiple planes are then intersected to produce an estimate of the 3D coordinate of the target. The commercially available ArcSecond system works in the same way.

Even though the sensors and sources have apparently swapped places in the CODA mxp30 and the Minnesota Scanner, they are both fundamentally making an angle measurement in the laboratory coordinate system. Thus, while the moving sensor on the Minnesota Scanner is “looking out”, it shares all the characteristics of a “outside-looking-in” system. What matters is the coordinate frame in which the measurements are made.

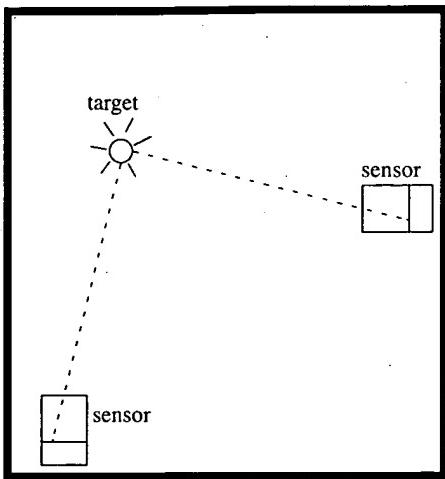


Figure 3.24: Two sensors are shown fixed in lab space. They observe a moving target. Here in ‘Flatland’ the position of the image on the 1D image-line of the sensor determines the equation of a line that passes through the target. The intersection of two of these lines determines the 2D position.

The positional sensitivity of the systems above is determined by the angular resolution of the optical sensors, the distance between target and sensor, and on the geometric configuration of the sensors. The angular resolution is determined by the field of view of the sensor and the resolution of measurements on the image plane. The designer of such a system always has a trade off between positional accuracy and working volume; higher accuracy requires a smaller working volume while larger working volume implies lower accuracy.

As shown in Figure 3.25 below, when the geometric configuration of the system is such that the sensors are physically close together compared to the distance to the target, the planes (or lines) they determine are nearly parallel. This results in near singularity of the resulting equations and poor positional accuracy in the direction aligned with the planes. This problem is aggravated by uncertainty in the sensor estimates. Rather than being planes or lines, the constraint provided by a single sensor is better modeled by a cone or a wedge. For maximum positional accuracy the sensors should be far apart and at nearly right angles to one another. Unfortunately placing the sensors far apart may give rise to line-of-sight problems because all sensors must have a clear view of each target.

In order to determine orientation, multiple targets must be arranged in a rigid configuration. Then the relative positions of the targets can be used to derive orientation. The sensitivity of the resulting system to small rotations is determined both by its positional sensitivity and the distance between targets. Orientation sensitivity is maximized by increasing the distance between targets but this can quickly result in a physically unwieldy device.

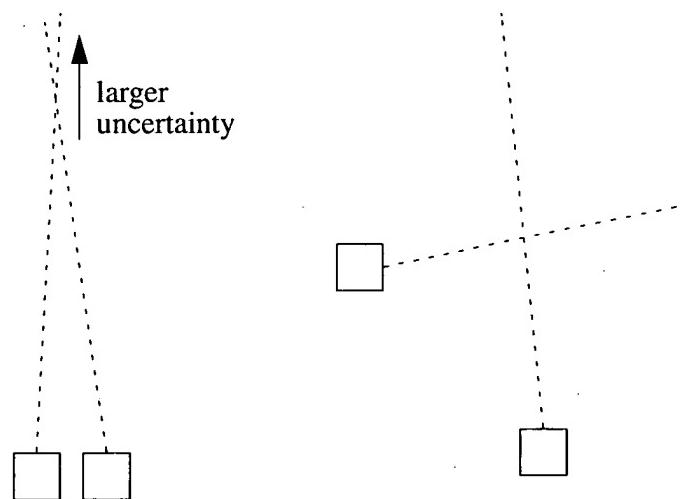


Figure 3.25: When sensors are close together as on the left, the equations of the lines they determine are nearly parallel resulting in less positional sensitivity in the direction parallel to the lines. When the sensors are far apart and at nearly right angles the sensitivity is greatest.

3.2.2 Measurements in the User's Frame

The HiBall Tracker [Welch, 1999 #307; Welch, 2001 #308] uses a golf-ball sized cluster of six 2D optical sensors looking out at LED beacons fixed in the environment. In the HiBall, angular measurements are made in the moving coordinate system of the user as shown in Figure 3.26, rather than in the fixed coordinate system of the lab.

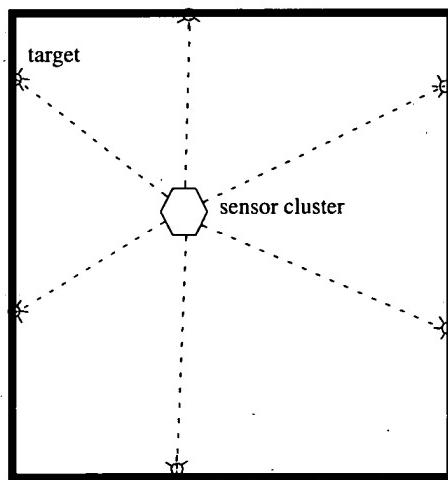


Figure 3.26: A cluster of six sensors observes targets that are at fixed locations in ‘Flatland’. Each observation determines a constraint between the position of the moving cluster and its orientation. A minimum of three observations are required to determine the position and orientation but to ensure uniqueness of the solution and to allow for measurement errors more are better.

Unlike the systems above for which one sighting determines a mathematically simple constraint such as a plane or line equation, sighting one target with one of the cameras in the cluster provides a difficult to visualize constraint on the relationship between the

cluster's position and its orientation. Given as few as three sightings it is possible to solve a nonlinear system of equations to determine the position and orientation of the cluster [Azuma, 1991 #260]. With only three sightings there are up to four solutions that can be arbitrarily close together [Fischler, 1981 #547]. Larger numbers of sightings provide a unique solution and also allow the use of least-squares estimation to reduce the effects of noise on the measurements.

The HiBall does not solve such nonlinear equations to determine its position. Instead it uses the SCAAT algorithm described in Section 4.2.5.

The orientation sensitivity of the HiBall is determined by the angular sensitivity of the optical sensors in the cluster. The HiBall achieves high angular sensitivity using very narrow fields of view (approximately 6 degrees) and high-resolution analog sensors (lateral-effect photo diodes [Wallmark, 1957 #548] with approximately 1 part in 2000 resolution). This requires that the room-mounted LEDs are densely packed to ensure that sufficient numbers are continuously visible.

The positional sensitivity of the HiBall is similar to the systems described above. It varies with distance to the target and with sensor resolution in the same way.

Comparing systems that measure in the laboratory frame with those that measure in the user's frame we see that user-centered measurements can provide higher orientation accuracy and comparable positional accuracy for systems of practical size. Systems with moving targets have the advantage that the targets are smaller and lighter and thus easier to attach to the user. Lab-based sensors are probably preferred when position is the only or most important measurement required.

3.3 Hybrid Systems

As described in Section 3.1, every type of sensor has fundamental limitations related to the associated physical medium. In addition there are practical limitations imposed by the measurement systems, and application-specific limitations related to the motion characteristics of the target being tracked. These limitations continuously affect the *quantity* and *quality* of the information. The result is that no single medium or sensor type provides the necessary performance over the wide spectrum of temporal and spatial characteristics desired for many applications. Happily several mediums exhibit complementary behavior, and these systems can be combined to leverage the strengths of each medium as needed. Systems that employ such mixed mediums are called *hybrid systems*.

A number of research and commercial groups have recognized that hybrid systems are necessary for some applications, and have constructed hybrids that combine multiple sensors including inertial, video, and GPS [Verplaetse, 1997 #300; Verplaetse, 1996 #234; Golding, 1999 #272; Azuma, 1998 #493; Azuma, 1999 #494; Azuma, 1999 #259; Azuma, 1994 #255; Azuma, 1995 #338; Behringer, 1999 #496; You, 1999 #311; You, 1999 #312; Pasman, 1999 #292; Foxlin, 1998 #270; Foxlin, 1994 #131; Foxlin, 1996 #509].

Inertial Tracking

One of the most popular technologies (mediums) used to improve or *augment* the performance of other mediums is to incorporate accelerometers and gyros in some form of an *inertial navigation system* as in [Emura, 1994 #78; List, 1983 #421; Azuma, 1995 #338; Azuma, 1994 #255; Azuma, 1999 #494; Azuma, 1999 #259]. The reason is that inertial navigation systems exhibit relatively low error at high frequencies and velocities, and are very responsive. This is due in part to the fundamental medium, and in part to the nature of inertial devices and our ability to sample their signals at a relatively high rate, typically on the order of thousands of samples per second. (Contrast this with magnetic, acoustic, and optical system can typically only be sampled hundreds of times per second.) Unfortunately as described in Section 3.1.2, they also exhibit high error at lower frequencies and velocities. At low velocities (very slow or no movement) one must in practice contend with pronounced bias and drift error (noise). As movement slows, such noise begins to grow with respect to the true signal, resulting in unbounded error growth.

Two examples of inertial hybrids are inertial-acoustic, and inertial-optical. The most well known example of the former is the commercial system described by [Foxlin, 1998 #270] and marketed by Intersense [Intersense, 2000 #277]. This system seeks to overcome the temporal and spatial shortcomings of purely acoustic systems (Section 3.1.1) by adding a relatively fast and robust inertial system. Again, the inertial system could not be used alone, but in combination with the acoustic system one can cover a wider spectrum of motion and performance. As a side note, the system described in [Foxlin, 1998 #270] actually uses *three* mediums—it uses pulsed infrared light to aid in the timing (triggering) of acoustic signals.

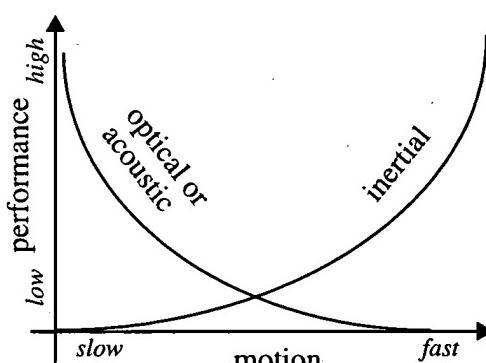


Figure 3.27: A qualitative comparison of inertial vs. optical or acoustic

A second example is an inertial-optical hybrid. As with any hybrid, the complementary behavior of each system is leveraged to obtain more accurate and stable tracking information than either system alone. With an inertial system, bias and drift errors dominate (grow unbounded) during periods of slow movement. However, during such periods the error can be controlled by an optical system, which would typically exhibit its best behavior under such conditions. Conversely an optical typically performs worst during very rapid movement,

precisely the conditions where the inertial signal-to-noise ratio is high (see Figure 3.10). In addition, while a typical vision-based optical system using multi-pixel cameras is affected by unrelated motion in an environment, an inertial system is *not* and can provide assistance with static visual feature discrimination. Figure 3.27 offers a qualitative comparison of the complementary relationship between the performance of inertial and optical (or acoustic) mediums.

4. Approaches

4.1 Traditional Closed-Form Approaches

We get the intuition from linear algebra that we need at least as many equations (or constraints) as there are unknowns to solve a system of equations. More equations might be required for a non-linear system. In this section we will examine a few solutions to particular tracking problems using this traditional approach. In Section 4.2.5 we examine a new approach that sequentially applies a single constraint at a time.

The key thing to notice is the variety of the mathematical approaches. These are only a few of the many formulations for these problems.

4.1.1 Range Trackers

For 3D position tracking using range common arrangements are three fixed microphones and a single moving source or three fixed sources and a single moving microphone. Mathematically, we must solve three simultaneous sphere equations where we know the origin (e.g. x_0, y_0, z_0) and the radius (e.g. r_0) for each sphere.

$$\begin{aligned}(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 &= r_0^2 \\(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= r_1^2 \\(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= r_2^2\end{aligned}\tag{4.1}$$

The solution of these equations is very complicated but we can simplify it greatly by aligning the three microphones with the axes of the coordinate system as follows:

- a. put microphone 0 at the origin of the coordinate system
- b. put microphone 1 out the X axis by one unit
- c. put microphone 2 out the Y axis by one unit
- d. all three microphones are in the Z = 0 plane.

This amounts to allowing the microphones to define the coordinate system rather than attempting to embed them in an existing coordinate system. We can, of course, get from these *microphone coordinates* to laboratory coordinates with a separate linear transform. The simplified equations are now:

$$\begin{aligned}x^2 + y^2 + z^2 &= r_0^2 \\(x - 1)^2 + y^2 + z^2 &= r_1^2 \\x^2 + (y - 1)^2 + z^2 &= r_2^2\end{aligned}\tag{4.2}$$

And their solution is:

$$\begin{aligned}x &= \frac{r_0^2 - r_1^2 + 1}{2} \\y &= \frac{r_0^2 - r_2^2 + 1}{2} \\z &= \pm \sqrt{r_0^2 - x^2 - y^2}\end{aligned}\tag{4.3}$$

Notice that the sign ambiguity on z prevents us from determining which side of the plane the target is on. This is a common problem with nonlinear systems of equations. The three spheres actually intersect in 2 points as shown in Figure 3.1 on page 30. Usually some design constraint is used to restrict z to the positive or negative subspace.

Another subtle point in this formulation and the others in this section is the assumption that the multiple measurements correspond to a single position of the target. For an acoustic tracker with fixed microphones and a moving source, this assumption is likely valid because the microphones respond to a single acoustic burst from the source. On the other hand, for a tracker with a moving microphone, this assumption is likely to be violated if the sources must be operated sequentially to avoid interference. In this case the microphone may have moved between successive measurements. Violating the assumption of simultaneous measurements results in measurement errors that vary with the target's rate of motion. See "The Simultaneity Assumption" on page 71 for further discussion.

4.1.2 Optical Trackers with Fixed 2D Sensors

Each camera in an optical system with 2D sensors that are fixed in laboratory coordinates determines a ray in 3D. The ray can be described using the parametric form with parameter s_1 for camera 1 and s_2 for camera 2. The parameters vary from 0 at the center of projection of a camera to infinity. The parametric equations are:

$$\begin{aligned}A_1 &= C_1 + s_1 D_1 \\A_2 &= C_2 + s_2 D_2\end{aligned}\tag{4.4}$$

C_1 and C_2 are the centers of projection of the cameras. D_1 is the unit-length direction vector determined by the image of the target on the image plane of camera 1. Likewise for D_2 . The baseline $B = C_2 - C_1$ is the vector between the centers of projection.

These two rays in three space almost certainly do not intersect. Of course the exact two lines must intersect because they result from images of the same point in space. But errors in calibration of the cameras and in determining the imaged coordinates in each camera will result in line equations that likely do not intersect.

We want to determine the point in 3D that is closest to both rays. We do this by finding the parameter values that minimize the distances between the lines. That is, we want to minimize

$$\|(C_2 + s_2 D_2) - (C_1 + s_1 D_1)\| \quad (4.5)$$

This equation is the length of the line joining the two rays. Since the shortest line joining the two rays must be perpendicular to each of the rays, it must be true that

$$\begin{aligned} [(C_2 + s_2 D_2) - (C_1 + s_1 D_1)] \cdot D_1 &= 0 \\ [(C_2 + s_2 D_2) - (C_1 + s_1 D_1)] \cdot D_2 &= 0 \end{aligned} \quad (4.6)$$

This system of two equations in two unknowns has the solution

$$\begin{aligned} s_1 &= \frac{(B \cdot D_1) - (D_2 \cdot D_1)(B \cdot D_2)}{1 - (D_1 \cdot D_2)^2} \\ s_2 &= \frac{(D_1 \cdot D_2)(B \cdot D_1) - (B \cdot D_2)}{1 - (D_1 \cdot D_2)^2} \end{aligned} \quad (4.7)$$

The point closest to the two rays is the midpoint of this shortest line segment

$$\tilde{P} = \frac{(C_1 + s_1 D_1) + (C_2 + s_2 D_2)}{2} \quad (4.8)$$

Examining equation (4.7) we can see the source of the difficulty described in Section 3.1.5 when the sensors are close together relative to the distance to the target. In this case the direction vectors D_1 and D_2 will be nearly parallel. Thus their dot product will be nearly one and the denominator of the equations will grow very small. This will amplify the effect of small errors.

4.1.3 Optical Trackers with Fixed 1D Sensors

The following approach may also be used with 2D sensors by considering that you are given four 1D measurements rather than two 2D measurements.

By a calibration process we determine the coefficients that map the 1D sensor coordinate to a plane in 3D. The plane for sensor i will produce an equation of the form

$$A_i x + B_i y + C_i z = D_i \quad (4.9)$$

Where $[x, y, z]^T$ is the 3D coordinate of the tracked point. With three such linear equations we can use standard solution techniques to solve for the position.

$$\begin{aligned} M &= \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{bmatrix} \\ M \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \quad (4.10) \\ \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= M^{-1} \cdot \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \end{aligned}$$

With more than three sensors we can use least-squares solution methods to determine the position with the minimum squared error.

$$\begin{aligned} M &= \begin{bmatrix} A_1 & B_1 & C_1 \\ \dots & \dots & \dots \\ A_n & B_n & C_n \end{bmatrix} \\ M \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= \begin{bmatrix} D_1 \\ \dots \\ D_n \end{bmatrix} \quad (4.11) \\ M^T \cdot M \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= M^T \cdot \begin{bmatrix} D_1 \\ \dots \\ D_n \end{bmatrix} \\ \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= (M^T \cdot M)^{-1} \cdot M^T \cdot \begin{bmatrix} D_1 \\ \dots \\ D_n \end{bmatrix} \end{aligned}$$

Analogously to the 2D case described earlier, if the target is far away relative to the distance between cameras, rows of the matrix M will be very similar resulting in near singularity and amplification of small errors.

4.1.4 Optical Trackers with Moving Sensors

Azuma and Ward [Azuma, 1991 #260] document the Space-Resection approach to solving for the position and orientation of the sensor cluster. This method was used in the first generation “Ceiling Tracker” at UNC. The HiBall tracker uses a much simpler method that will be described later.

Their method is too complicated to describe here. They set up the system of non-linear equations that described the relationships among the known 3D coordinates of the LEDs, the known 2D image-plane coordinates for the sightings of the LEDs, and the unknown position and orientation of the camera cluster. They solved the non-linear system of equations using an iterative approach that required a good initial guess. During normal system operation the previously known pose was usually an excellent guess for the current pose and the iterative method converged rapidly. Initialization at system startup was accomplished by sequentially trying a small set of different orientations to see if any will converge to a likely solution. The convergence region of the algorithm was large enough that it could acquire the initial position within a few seconds when the tracker was held upright at about head height.

4.2 Stochastic Approaches

While there are many application-specific approaches to “computing” (estimating) the position and orientation or *pose* of an object (see Section 4.1), most of these methods do not inherently take into consideration the noisy nature of the sensor measurements. While the requirements for the pose information varies with application, the fundamental source of information is the same: pose estimates are derived from *noisy* electrical measurements of mechanical, inertial, optical, acoustic, or magnetic sensors. This noise is typically statistical in nature (or can be effectively modeled as such), which leads us to *stochastic* methods for addressing the problems. Here we provide a very basic introduction to the subject, primarily aimed at preparing the reader for the material in the appendices. For a more extensive discussion of stochastic estimation see for example [Kailath, 2000 #279; Lewis, 1986 #89].

4.2.1 State-Space Models

State-space models are essentially a notational convenience for estimation and control problems, developed to make what would otherwise be a notationally-intractable analysis tractable. Consider a dynamic process described by an n -th order difference equation (similarly a differential equation) of the form

$$y_{i+1} = a_{0,i}y_i + \dots + a_{n-1,i}y_{i-n+1} + w_i, \quad i \geq 0,$$

where $\{w_i\}$ is a *zero-mean* (statistically) *white* (spectrally) random “noise” process with autocorrelation

$$E(w_i w_j) = Q_i \delta_{ij},$$

and initial values $\{y_0, y_{-1}, \dots, y_{-n+1}\}$ are zero-mean random variables with a known $n \times n$ covariance matrix

$$P_0 = E(y_{-j} y_{-k}), j, k \in \{0, n-1\}.$$

Also assume that

$$E(w_i y_i) = 0 \text{ for } -n+1 \leq i \leq 0 \text{ and } i \geq 0,$$

which ensures [Kailath, 2000 #279] that

$$E(w_i y_i) = 0, i \geq j \geq 0.$$

In other words, that the noise is statistically independent from the process to be estimated. Under some other basic conditions [Kailath, 2000 #279] this difference equation can be re-written as

$$\hat{x}_{i+1} = \underbrace{\begin{bmatrix} y_{i+1} \\ y_i \\ y_{i-1} \\ \vdots \\ y_{i-n+2} \end{bmatrix}}_{A} + \underbrace{\begin{bmatrix} a_0 & a_1 & \dots & a_{n-2} & a_{n-1} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}}_{\hat{x}_i} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}}_G w_i$$

which leads to the *state-space model*

$$\hat{x}_{i+1} = A \hat{x}_i + G w_i$$

$$\hat{y}_i = [1 \ 0 \ \dots \ 0] \hat{x}_i$$

or the more general form

$$\hat{x}_{i+1} = A \hat{x}_i + G w_i \quad (4.12)$$

$$\hat{y}_i = H_i \hat{x}_i \quad (4.13)$$

Equation (4.12) represents the way a new state \hat{x}_{i+1} is modeled as a linear combination of both the previous state \hat{x}_i and some *process noise* w_i . Equation (4.13) describes the way the process measurements or *observations* \hat{y}_i are derived from the internal state \hat{x}_i .

These two equations are often referred to respectively as the *process model* and the *measurement model*, and they serve as the basis for virtually all linear estimation methods, such as the *Kalman filter* described below.

4.2.2 The Observer Design Problem

There is a related general problem in the area of linear systems theory generally called the *observer design problem*. The basic problem is to determine (estimate) the internal *states* of a linear system, given access only to the system's *outputs*.¹ This is akin to what people often think of as the “black box” problem where you have access to some signals coming from the box (the outputs) but you cannot directly observe what's inside.

The many approaches to this basic problem are typically based on the state-space model presented in the previous section. There is typically a *process model* that models the transformation of the process state. This can usually be represented as a linear stochastic difference equation similar to equation (4.12):

$$x_k = Ax_{k-1} + Gw_k. \quad (4.14)$$

In addition there is some form of *measurement model* that describes the relationship between the process state and the measurements. This can usually be represented with a linear expression similar to equation (4.13):

$$z_k = Hx_k + v_k. \quad (4.15)$$

The terms w_k and v_k are random variables (see Section 2.2.1 on page 21) representing the process and measurement noise respectively.

Measurement and Process Noise

There are many sources of noise in sensor measurements. For example, each type of sensor has fundamental limitations related to the associated physical medium, and when pushing the envelope of these limitations the signals are typically degraded. In addition, some amount of random electrical noise is added to the signal via the sensor and the electrical circuits. The time-varying ratio of “pure” signal to the electrical noise continuously affects the *quantity* and *quality* of the information. The result is that information obtained from any one sensor must be qualified as it is interpreted as part of an overall sequence of pose estimates, and analytical measurement models typically incorporate some notion of random measurement noise or uncertainty v_k as shown above in equation (4.15).

1. Access to the system's control inputs is also presumed, but less relevant in the case of tracking and motion capture, so we will omit that aspect. See for example [Kailath, 2000 #279] for more information.

In the case of tracking or motion capture of humans, there is the additional problem that the user's intended motion is essentially completely unknown. While we can make predictions over relatively short intervals using models based on recent motion as a guide (see Section 5.3), such predictions assume that the user's motion is predictable, which is not always the case. The result is that like sensor information, ongoing estimates of the user pose must be qualified as they are combined with measurements in an overall sequence of pose estimates. In addition, analytical motion or process models typically incorporate some notion of random motion or uncertainty w_k as shown above in equation (4.14).

4.2.3 Optimal Estimation—The Kalman Filter

Among the substantial number of mathematical tools that can be used for stochastic pose estimation from noisy sensor measurements, one of the most well-known and often-used tools is what is known as the *Kalman filter*. The Kalman filter is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-data linear filtering problem [Kalman, 1960 #87].

The filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is *optimal* in the sense that it minimizes the estimated *error* covariance—when some presumed conditions are met. Since the time of its introduction, the *Kalman filter* has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. This is likely due in large part to advances in digital computing that made the use of the filter practical, but also to the relative simplicity and robust nature of the filter itself. Rarely do the conditions necessary for optimality actually exist, and yet the filter apparently works well for many applications in spite of this situation.

Of particular note here, the Kalman filter has been used extensively for tracking in interactive computer graphics. We use a *single-constraint-at-a-time* Kalman filter (see page 66) in our HiBall Tracking System [Welch, 2001 #308; Welch, 1999 #307] which is commercially available from 3rdTech [3rdTech, 2000 #153]. It has also been used for motion prediction [Azuma, 1995 #64; Azuma, 1994 #255], and it is used for multi-sensor (inertial-acoustic) fusion in the commercial Constellation™ wide-area tracking system by Intersense [Foxlin, 1998 #270; Intersense, 2000 #277]. See also [Azarbayejani, 1994 #336; Emura, 1994 #78; Emura, 1994 #505; Fuchs (Foxlin), 1993 #271; Mazuryk, 1995 #284; Van Pabst, 1993 #535].

We maintain a popular web site on the topic of the Kalman filter. The web address is

<http://www.cs.unc.edu/~welch/kalman/>.

On this site you will find references to (and some copies of) introductory and advanced material on the Kalman filter. New for 2001—we will be bringing on-line a Java-based *Kalman Filter Learning Tool*. In addition, we are also teaching an Introduction to the Kalman Filter two-hour tutorial at SIGGRAPH 2001 (Course 8), for which there are separate course notes. Finally, in Appendix A of this course pack (page 79) we have

included a copy of our own introductory technical report. Beyond this Appendix, a very “friendly” introduction to the general idea of the Kalman filter can be found in Chapter 1 of [Maybeck, 1979 #93]—which is available from the above Kalman filter web site, while a more complete introductory discussion can be found in [Sorenson, 1970 #101], which also contains some interesting historical narrative. More extensive references include [Gelb, 1974 #82; Grewal, 2001 #245; Maybeck, 1979 #93; Lewis, 1986 #89; Jacobs, 1993 #86; Brown, 1996 #246].

4.2.4 Hybrid or Multi-Sensor Fusion

Stochastic estimation tools such as the Kalman filter (see Appendix A on page 79) can be used to combine or *fuse* information from different mediums or sensors for *hybrid systems* (see Section 3.3 on page 54). The basic idea is to use the Kalman filter to weight the different mediums most heavily in the circumstances where they each perform best, thus providing more accurate and stable estimates than a system based on any one medium alone. In particular, the *indirect feedback* Kalman filter shown in Figure 4.1 (also called a *complementary* or *error-state* Kalman filter) is often used to combine the two mediums [Maybeck, 1979 #93]. In such a configuration, the Kalman filter is used to estimate the *difference* between the current inertial and optical (or acoustic) outputs, i.e. it continually estimates the *error* in the inertial estimates by using the optical system as a second (redundant) reference. This error estimate is then used to correct the inertial estimates. The adjustment or *tuning* of the Kalman filter parameters then determines the weight of the correction as a function of frequency. By slightly modifying the Kalman filter, adaptive velocity response can be incorporated also. This can be accomplished by adjusting (in real time) the expected optical measurement error as a function of the magnitude of velocity. The dashed line in Figure 4.1 indicates the additional use of inertial estimates to help a image-based optical system to prevent tracking of moving scene objects (i.e. unrelated motion in the environment).

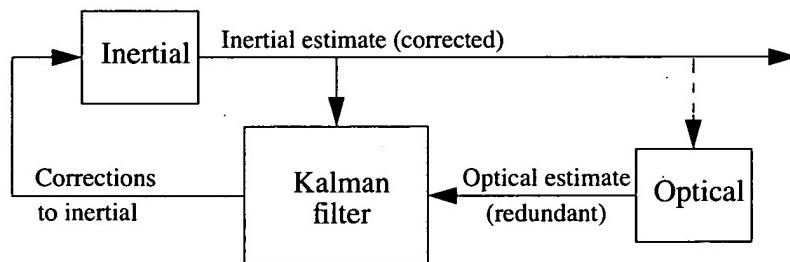


Figure 4.1: The Kalman filter used in an *indirect-feedback* configuration to optimally weight inertial and optical information.

In such a configuration, the Kalman filter uses a common *process model*, but a distinct *measurement model* for each of the inertial and optical subsystems. See Appendix A on page 79 for more information about these models.

4.2.5 Single-Constraint-at-a-Time Tracking

A conventional approach to pose estimation is to collect a group of sensor measurements and then to attempt to simultaneously solve a system of equations that together completely constrain the solution. For example, the 1991 UNC-Chapel Hill wide-area opto-electronic tracking system [Ward, 1992 #540; Wang, 1990 #301] collected a group of diverse measurements for a variety of LEDs and sensors, and then used a method of simultaneous non-linear equations called *Collinearity* to estimate the pose of the head-worn sensor fixture [Azuma, 1991 #260]. There was one equation for each measurement, expressing the constraint that a ray from the front principal point of the sensor lens to the LED, must be collinear with a ray from the rear principal point to the intersection with the sensor. Each estimate made use of typically 20 (or more) measurements that together over-constrained the solution.

This *multiple constraint* method had several drawbacks. First, it had a significantly lower estimate rate due to the need to collect multiple measurements per estimate. Second, the system of non-linear equations did not account for the fact that the sensor fixture continued to move throughout the collection of the sequence of measurements. Instead the method effectively assumes that the measurements were taken simultaneously. The violation of this *simultaneity assumption* could introduce significant error during even moderate motion. Finally, the method provided no means to identify or handle unusually noisy individual measurements. Thus, a single erroneous measurement could cause an estimate to jump away from an otherwise smooth track.

In contrast, there is typically nothing about solutions to the observer design problem in general (Section 4.2.2), or the Kalman filter in particular (Section 4.2.3), that dictates the ordering of measurement information. In 1996 we introduced a new Kalman filter-based approach to tracking, an approach that exploits this flexibility in measurement processing. The basic idea is to update the pose estimate as each new measurement is made, rather than waiting to form a complete collection of measurement. Because single measurements under-constrain the mathematical solution, we refer to the approach as *single-constraint-at-a-time* or SCAAT tracking [Welch, 1997 #304; Welch, 1996 #303]. The key is that the single measurements provide *some* information about the tracker state, and thus can be used to incrementally improve a previous estimate. We intentionally fuse each individual “insufficient” measurement immediately as it is obtained. With this approach we are able to generate estimates more frequently, with less latency, with improved accuracy, and we are able to estimate the LED positions on-line concurrently while tracking. This approach is used in our laboratory-based HiBall Tracking System [Welch, 2001 #308; Welch, 1999 #307], the commercial version of the same system [3rdTech, 2000 #153], and the commercial systems manufactured by Intersense [Foxlin, 1998 #270; Intersense, 2000 #277].

One of the most interesting things about the SCAAT approach is that it can be almost universally applied in place of conventional approaches (Section 4.1). Essentially all you need is to be able to predict a sensor measurement given a current pose estimate. In other words, if you can formulate a measurement model as in equation (4.15) in Section 4.2.2,

you can use the SCAAT approach. In fact, one of the unusual things about the approach is that you never actually compute the pose directly, you only compute the measurement you think the sensor should “see.” For computer graphics people, the measurement model for optical sensors in particular is “simple” as it typically resembles the normal graphics viewing transformations.

Consider for a moment the UNC hybrid landmark-magnetic tracker presented at SIGGRAPH 96 [State, 1996 #469]. This system uses an off-the-shelf Ascension magnetic tracking system along with a vision-based landmark recognition system to achieve superior synthetic and real image registration for augmented reality assisted medical procedures. The vision-based component attempts to identify and locate multiple known landmarks in a single image before applying a correction to the magnetic readings. A SCAAT implementation would instead predict the location of a landmark in the image, then identify and locate that single landmark in the actual image. It would process one landmark per image update in this fashion. Not only would this approach increase the frequency of landmark-based correction (given the necessary image processing) but it would offer the added benefit that unlike the implementation presented in [State, 1996 #469], *no special processing would be needed* for the cases where the number of visible landmarks falls below the number necessary to determine a complete position and orientation solution. The SCAAT implementation would simply cycle through any available landmarks, one at a time. Even with only one visible landmark the method would continue to operate as usual, using the information provided by the landmark sighting to refine the estimate.

For more information see [Welch, 1997 #304] and [Welch, 1996 #303], the latter of which is included at the end of this course pack in Appendix C.

5. Problems and Insights

In this chapter we consider some of the primary sources of error in estimates from tracking and motion capture systems. While we focus specifically on head tracking for interactive computer graphics, the basic principles are applicable to tracking of hands, and even to motion capture. We attempt to provide some insight into the source of the error, and even a means for addressing (not necessarily *solving*) one of the biggest problems: end-to-end delay in the entire tracking and graphics pipeline.

5.1 Classification of Error

There are of course many causes of visual error in interactive computer graphics systems. There are many people (aside from the authors) who would argue that various errors originating in the tracking system dominate all other sources. In his 1995 Ph.D. dissertation thoroughly analyzing the sources of error in an Augmented Reality system for computer-aided surgery, Rich Holloway stated

Clearly, the head tracker is the major cause of registration error in AR systems. The errors come as a result of errors in aligning the tracker origin with respect to the World CS (which may be avoidable), measurement errors in both calibrated and multibranched trackers, and delay in propagating the information reported by the tracker through the system in a timely fashion.

Rich Holloway's dissertation offers a very thorough look at the sources of error in the entire AR pipeline, including the stages associated with tracking. Much of the dissertation is applicable to VR systems in general, and even motion capture. We highly encourage you to take a look if you are really interested in a rigorous mathematical analysis. Chapter 8 of the dissertation discusses some methods for combating the problems introduced by tracker error, in particular delay. The dissertation is available from <http://www.cs.unc.edu/Publications/Dissertations.html>.

Sources of Error

For a person designing, calibrating, or using a tracking or motion capture system, it is useful to have some insight into where errors come from. As [Deering, 1992 #552] notes, "...the visual effect of many of the errors is frustratingly similar." This is especially true for tracking errors. We have seen people build VR applications with obvious head tracker

transformation errors, and yet people had great difficulty figuring out what part of the long sequence of transforms was wrong, if it was a static calibration error, or a simple sign error.

Yet even when all of the transforms are of the correct form, the units of translation and orientation match, and all the signs are correct, there are still unavoidable errors in motion tracking, errors that confound even the most experienced of practitioners of interactive computer graphics. No matter what the approach (see Chapter 4), the process of pose estimation can be thought of as a sequence of events and operations. The sequence begins with the user motion, and typically ends with a pose estimate arriving at the host computer, ready to be consumed by the application. Clearly by the time a pose estimate arrives at the host computer it is already “late”—and you still have to render an image and wait for it to be displayed! Section 5.3 offers some hope for addressing the long delays and in some sense “catching up” with the user motion, but that doesn’t mean that we don’t want to minimize the delay, and to understand how all of the various errors affect the outcome.

The sources of error in tracking and motion capture can generally be divided into two primary classes. The first includes all errors related to making static measurements, either off line prior to running an application, or on line during normal operation. We call this *static measurement error*. The second includes all errors that arise from the inevitable sources of delay in the tracking pipeline. We call this *delay-induced error*.

5.1.1 Static Measurement Error

Static Field Distortion

For an *immobile* sensor (static motion), we can divide the measurement errors into two types: *repeatable* and *nonrepeatable*. Some trackers (for example, magnetic ones) have systematic, repeatable distortions of their measurement volume which cause them to give erroneous data; we will call this effect *static field distortion*. The fact that these measurement errors are repeatable means that they can be measured and corrected as long as they remain unchanged between this calibration procedure and run time.

Random Noise or Jitter

Here we consider the *non-repeatable* errors made by the tracker for an *immobile* sensor. As we discussed in general in Section 2.1.1, some amount of noise in the sensor inputs is inevitable with any measurement system, and this measurement noise typically leads to random noise or *jitter* in the pose estimates. By our definition, this type of error is not repeatable and therefore not correctable *a priori* via calibration. Moreover, the amount of jitter in the tracker’s outputs limits the degree to which the tracker can be calibrated. The amount of jitter is often proportional to the distance between the sensor(s) and the source(s), and may become relatively large near the edge of the tracker’s working volume.

5.1.2 Delay-Induced Error

Any measurement of a non-repeating, time-varying phenomenon is valid (at best) at the instant the sample occurs—or over the brief interval it occurs, and then becomes “stale” with the passage of time until the next measurement. The age of the data is thus one factor in its accuracy. Any delay between the time the measurement is made and the time that measurement is manifested by the system in a pose estimate contributes to the age and therefore the inaccuracy of that measurement. The older the tracker data is, the more likely that the displayed image will be misaligned with the real world.

We feel that concerns related to *dynamic error* (including *dynamic tracker error* and *delay-induced error* from above) deserve distinct discussion. This class of error is often less obvious when it occurs (you know something isn’t correct, but you don’t know why), and when you do recognize it, it is difficult to know where to look to minimize the effects.

First-Order Dynamic Error

Probably the most obvious effect here is the overall *dynamic error* caused by continued user motion after a tracker cycle (sample, estimate, produce) has started. If the user’s head is rotating with an angular velocity of $\dot{\theta}$ and translating with a linear velocity of \dot{x} then simple first-order models for the delay-induced orientation and translation error are given by

$$\epsilon_{\text{dyn}, \theta} = \dot{\theta}\Delta t \quad (5.1)$$

$$\epsilon_{\text{dyn}, x} = \dot{x}\Delta t \quad (5.2)$$

where Δt is the sum of the total motion delay Δt_m for the tracking system as described below in Section 5.2, as well as Δt_g , the delay through the remainder of the graphics pipeline—including rendering and image generation, video synchronization delay, frame synchronization delay, and internal display delay. The *video synchronization delay* is the amount of time spent waiting for a frame buffer to swap—on average 1/2 the frame time. (Synchronization delay in general is described more below.) The *internal display delay* is any delay added by the display device beyond the normal frame delay. For example, some LCD and DLP devices buffer images internally in a non-intuitive manner. The delay must be measured on a per-device basis if it is important.

The Simultaneity Assumption

Many popular tracking systems collect sensor measurements sequentially, and then assume (mathematically) that they were collected simultaneously. We refer to this as the *simultaneity assumption*. If the target remains motionless this assumption introduces no error. However if the target is moving, the violation of the assumption introduces error. Consider that typical arm and wrist motion can occur in as little as 1/2 second, with typical “fast” wrist tangential motion occurring at three meters per second [Atkeson, 1985 #556]. For the a typical magnetic tracker with 20-80 ms of latency, such “fast” motion

corresponds to approximately one to ten centimeters of translation *throughout* the sequence of sensor samples used for a single estimate. For systems that attempt sub-millimeter accuracies, even slow motion occurring during a sequence of sequential samples impacts the accuracy of the estimates. For example, in a multiple-sample system with $\Delta t_s = 30$ [ms] of total sample time, motion of only three centimeters per second corresponds to approximately one millimeter of target translation throughout the sequence of samples for one estimate. Figure 5.1 presents the results of a simulation from [Welch, 1996 #303] (page 161) which includes a more extensive analysis of this error source. Figure 5.1 shows how estimates can be pulled away from the truth by an error amount of ϵ_{sa} as the simultaneity assumption is violated.

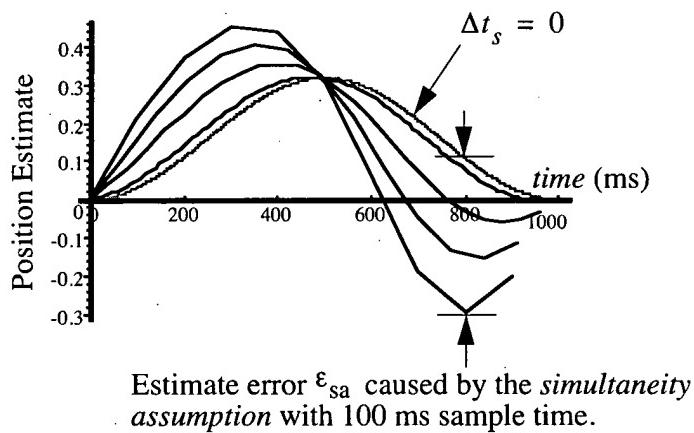


Figure 5.1: Simulated error resulting from the simultaneity assumption. The family of curves shows how simulated position estimates become skewed by the simultaneity assumption as a target undergoes motion with a one Hertz sinusoidal velocity. Note the increasing skew of the estimates with total sensor sample times of $\Delta t_s \in \{0, 10, 40, 70, 100\}$ ms. Details appear in [Welch, 1996 #303].

Sensor Sample Rate

Per Shannon's sampling theorem [Jacobs, 1993 #86] the measurement or *sampling rate* r_{ss} should be at least twice the true target motion bandwidth, or an estimator may track an alias of the true motion. Given that common arm and head motion bandwidth specifications range from 2 to 20 Hz [Fischer, 1990 #555; Foxlin, 1993 #372; Neilson, 1972 #554], the *sampling rate* should ideally be greater than 40 Hz. Furthermore, the *estimation rate* r_e should be as high as possible so that slight (expected and acceptable) estimation error can be discriminated from the unusual error that might be observed during times of significant target dynamics.

Synchronization Delay

While other latencies (delays) certainly do exist in the typical VE system [Council, 1994 #358; Wloka, 1995 #557; Mine, 1993 #433] tracker latency is unique in that it determines how much time elapses before the first possible opportunity to respond to user motion. When the user moves, we want to know as soon as possible. Within the tracking system pipeline of events (and throughout the rendering pipeline) there are both fixed latencies associated with well-defined tasks such as executing functions to compute the pose, and variable latencies associated with the synchronization between well-defined asynchronous tasks. The latter is often called *synchronization delay*, although sometimes also *phase delay* or *rendezvous delay*. See for example Figure 5.2.

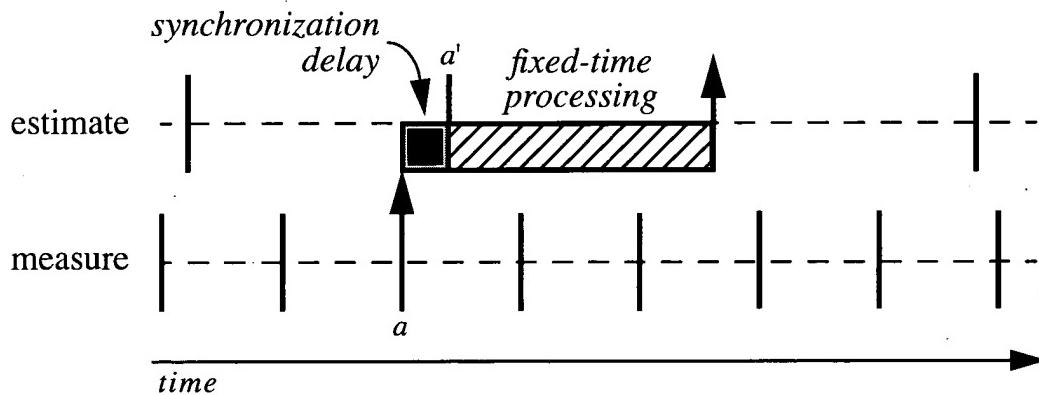


Figure 5.2: Synchronization delay. A measurement is taken at a but not used to estimate the pose until a' . The intervening time is called *synchronization delay*.

In the example of Figure 5.2, measurements and pose estimates occur at regular but *different* rates. Inevitably, any measurement will sit for some time before being used in to compute a pose estimate. At best, the measurement will be read immediately *after* it is made. At worst the measurement will be read just *before* it is replaced with a newer measurement. On average the delay would be $1/2$ the measurement rate.

5.2 Total Tracker Error

Figure 5.3 presents a more involved example, a sequence of inter-tracker events and the corresponding delays. Consider an instantaneous step-like user motion as depicted in Figure 5.3. The sequence of events begins at t_m , the instant the user begins to move. In this example the sensors are sampled at a regular rate $r_{ss} = 1/\tau_{ss}$, such as would typically be the case with video or a high-speed A/D conversion. On average there will be $\Delta t_{ss} = \tau_{ss}/2$ seconds of sample synchronization delay before any sample is used for pose estimation. Because the pose estimate computations are repeated asynchronously at the regular rate of $r_e = 1/\tau_e$ there will be an average of $\Delta t_e = \tau_e/2$ seconds of estimation synchronization delay, after which time the estimation will take τ_e seconds.

Assuming a client-server architecture such as [VRPN, 2001 #558] the final estimate will be written to a server communications buffer where it is being read at a rate of $r_{srb} = 1/\tau_{srb}$, and will therefore wait an average of $\Delta t_{srb} = \tau_{srb}/2$ seconds before being read and transmitted over the network to the client. The network transmission itself will take τ_{net} , and the final client read-buffer synchronization delay will take $\Delta t_{crb} = \tau_{crb}/2$ seconds, where $\tau_{crb} = 1/r_{crb}$ (the client read-buffer rate). The total (average) motion delay in this example is then

$$\begin{aligned}\Delta t_m &= t_m' - t_m \\ &= \Delta t_{ss} + \Delta t_e + \tau_e + \Delta t_{srb} + \tau_{net} + \Delta t_{crb} \\ &= \frac{1}{2r_{ss}} + \frac{1}{2r_e} + \tau_e + \frac{1}{2r_{srb}} + \tau_{net} + \frac{1}{2r_{crb}}\end{aligned}\quad (5.3)$$

where r_{ss} is the sensor sample rate, r_e is the estimate rate, $\tau_e = 1/r_e$, r_{srb} is the server read-buffer rate, τ_{net} is the network transmission time, and r_{crb} is the client read-buffer rate.

Note that this bound does not include any latency inherently added by pose estimate computations that also implement some form of filtering.

Summing the *static measurement error* from Section 5.1.1, the error ε_{sa} caused by violation of the simultaneity assumption, and the *dynamic error* given by equations (5.1) and (5.2), we get a total error of

$$\begin{aligned}\varepsilon_\theta &\approx \varepsilon_{stat, \theta} + \varepsilon_{sa, \theta} + \dot{\theta}(\Delta t_m + \Delta t_g) \\ \varepsilon_x &\approx \varepsilon_{stat, x} + \varepsilon_{sa, x} + \dot{x}(\Delta t_m + \Delta t_g)\end{aligned}\quad (5.4)$$

where Δt_m is from equation (5.3), and Δt_g includes the remainder of the graphics pipeline delay as described in “First-Order Dynamic Error” above in Section 5.1.2. Clearly the final rotation and translation error is sensitive to both the user motion velocity, and the total delay of the tracker and graphics pipeline.

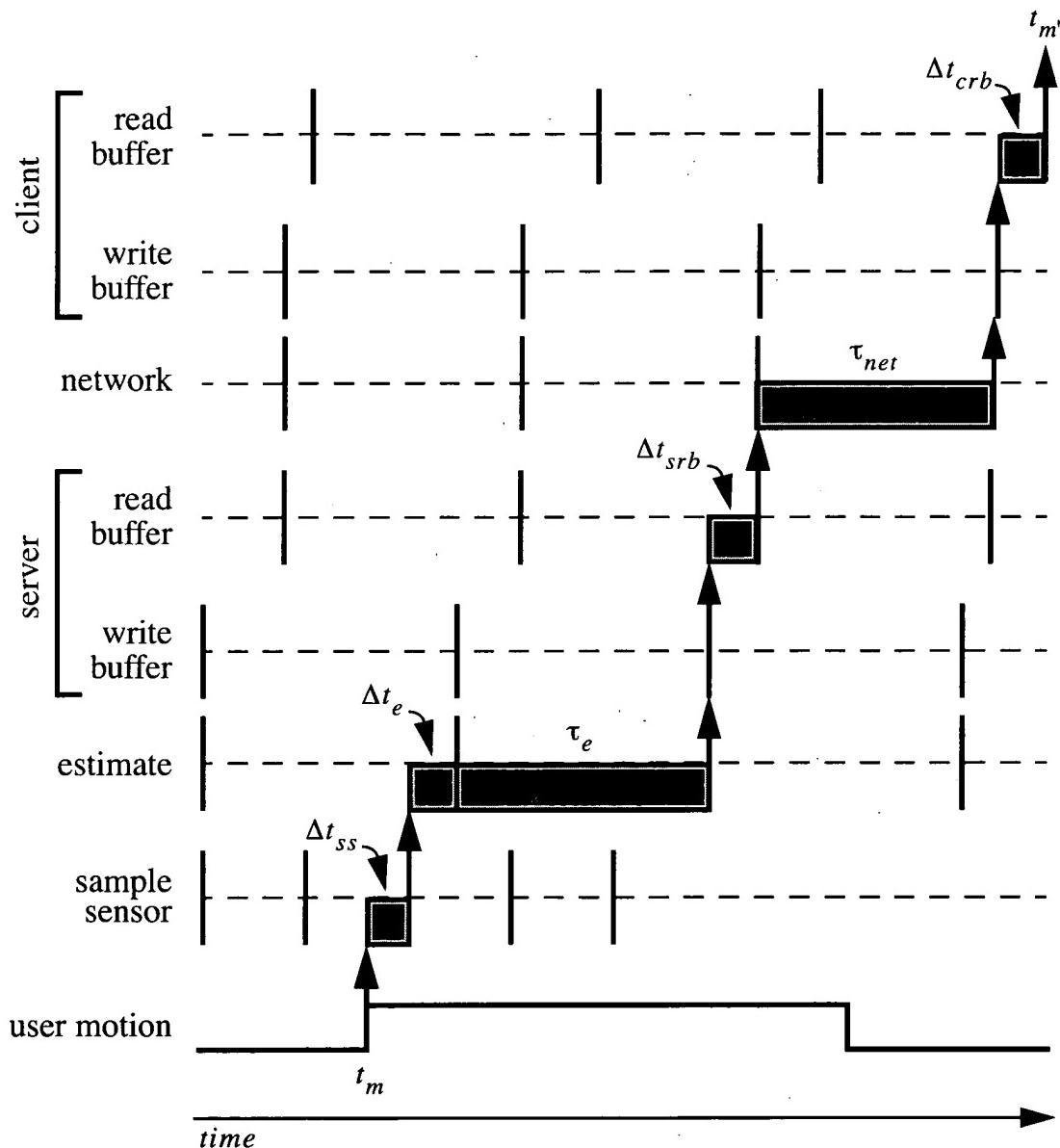


Figure 5.3: An example sequence of inter-tracker events and delays.

5.3 Motion Prediction

When trackers are used to implement VE or AR systems, end-to-end delays the total system will result in perceived *swimming* of the virtual world whenever the user's head moves. The delay causes the virtual objects to appear to follow the user's head motion with a velocity dependent error.

The sequence of events in a head-mounted display system goes something like this:

Time	Event
t_0	tracker measures user's pose
t_1	tracker reports the pose
t_2	application receives the reported pose
t_3	updated image is ready in the hidden buffer of a double-buffered display
t_4	buffer swap happens at vertical interval
t_5	image is scanned out to the display

Table 5.1: Time series of events in a head-mounted display system.

The interval from t_0 to t_5 is on the order of 30ms in the fastest systems and upwards to 200ms in the slowest. If the user is moving during this interval the image finally displayed at t_5 will not be appropriate for the user's new position. We are displaying images appropriate for where the user *was* rather than for where he *is*.

The most important step in combating this swimming is to reduce the end-to-end delay. This process can be taken only so far though. Each of the steps takes *some* time and this time is not likely to be reduced to negligible simply by accelerating the hardware.

After the avoidable delays have been eliminated we can mitigate the effect of the unavoidable delays by using motion prediction. Our goal is to extrapolate the user's past motion to predict where he will be looking at the time the new image is ready. As Azuma [Azuma, 1995 #64] points out, this is akin to driving a car by looking only the rear-view mirror. To keep the car on the road, the driver must predict where the road will go, based solely on the view of the past and knowledge of roads in general. The difficulty of this task depends on how fast the car is going and on the shape of the road. If the road is straight and remains so, then the task is easy. If the road twists and turns unpredictably, the task will be impossible.

Motion predictors attempt to extract information from past measurements to predict future measurements. Most methods, at their core, attempt to estimate the local derivatives so that a Taylor series can be evaluated to estimate the future value. The differences among methods are mostly in the type and amount of smoothing applied to the data in estimating the derivatives.

The simplest approach simply extends a line through the previous two measurements to the time of the prediction. This approach will be very sensitive to noise in the measurements. More sophisticated approaches will take weighted combinations of several previous measurements. This will reduce sensitivity to noise but will incur a delay in responding to rapid changes. All methods based solely on past measurements of position and orientation will face a trade off between noise and responsiveness.

Performance of the predictor can be improved considerably if direct measurements of the derivatives of motion are available from inertial sensors. As described earlier, linear accelerometers and rate gyros provide estimates of the derivatives of motion with high bandwidth and good accuracy. Direct measurements are superior to differentiating the position and orientation estimates because they are less noisy and are not delayed.

Azuma [Azuma, 1994 #255] demonstrated prediction using inertial sensors that reduced swimming in an augmented reality system by a factor of 5 to 10 with end-to-end delay of 80 [ms]. Further in [Azuma, 1995 #256] he shows that error in predictions based on derivatives and simple models of motion are related to the square of the product of the prediction interval and the bandwidth of the motion sequence. Doubling the prediction interval for the same sort in input will quadruple the error.

A. An Introduction to the Kalman Filter

This appendix is a copy of UNC technical report TR 95-041, written by Welch and Bishop in 1995. It is included to provide a ready and accessible introduction to both the discrete Kalman filter and the extended Kalman filter. This report and other useful material can be found at the authors' Kalman filter web site, <http://www.cs.unc.edu/~welch/kalman/>.

A.1 The Discrete Kalman Filter

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem [Kalman60]. Since that time, due in large part to advances in digital computing, the *Kalman filter* has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. A very “friendly” introduction to the general idea of the Kalman filter can be found in Chapter 1 of [Maybeck79], while a more complete introductory discussion can be found in [Sorenson70], which also contains some interesting historical narrative. More extensive references include [Gelb74; Grewal93; Maybeck79; Lewis86; Brown92; Jacobs93].

A.1.1 The Process to be Estimated

The Kalman filter addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}, \quad (\text{A.1})$$

with a measurement $z \in \Re^m$ that is

$$z_k = Hx_k + v_k. \quad (\text{A.2})$$

The random variables w_k and v_k represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions

$$p(w) \sim N(0, Q), \quad (\text{A.3})$$

$$p(v) \sim N(0, R). \quad (\text{A.4})$$

In practice, the *process noise covariance* Q and *measurement noise covariance* R matrices might change with each time step or measurement, however here we assume they are constant.

The $n \times n$ matrix A in the difference equation equation (A.1) relates the state at the previous time step $k - 1$ to the state at the current step k , in the absence of either a driving function or process noise. Note that in practice A might change with each time step, but here we assume it is constant. The $n \times l$ matrix B relates the optional control input $u \in \Re^l$ to the state x . The $m \times n$ matrix H in the measurement equation equation (A.2) relates the state to the measurement z_k . In practice H might change with each time step or measurement, but here we assume it is constant.

A.1.2 The Computational Origins of the Filter

We define $\hat{x}_k^- \in \Re^n$ (note the “super minus”) to be our *a priori* state estimate at step k given knowledge of the process prior to step k , and $\hat{x}_k \in \Re^n$ to be our *a posteriori* state estimate at step k given measurement z_k . We can then define *a priori* and *a posteriori* estimate errors as

$$\begin{aligned} e_k^- &= x_k - \hat{x}_k^-, \text{ and} \\ e_k &= x_k - \hat{x}_k. \end{aligned}$$

The *a priori* estimate error covariance is then

$$P_k^- = E[e_k^- e_k^{T^-}], \quad (\text{A.5})$$

and the *a posteriori* estimate error covariance is

$$P_k = E[e_k e_k^T]. \quad (\text{A.6})$$

In deriving the equations for the Kalman filter, we begin with the goal of finding an equation that computes an *a posteriori* state estimate \hat{x}_k as a linear combination of an *a priori* estimate \hat{x}_k^- and a weighted difference between an actual measurement z_k and a measurement prediction $H\hat{x}_k^-$ as shown below in equation (A.7). Some justification for equation (A.7) is given in “The Probabilistic Origins of the Filter” found below.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (\text{A.7})$$

The difference $(z_k - H\hat{x}_k^-)$ in equation (A.7) is called the measurement *innovation*, or the *residual*. The residual reflects the discrepancy between the predicted measurement $H\hat{x}_k^-$ and the actual measurement z_k . A residual of zero means that the two are in complete agreement.

The $n \times m$ matrix K in equation (A.7) is chosen to be the *gain* or *blending factor* that minimizes the *a posteriori* error covariance equation (A.6). This minimization can be accomplished by first substituting equation (A.7) into the above definition for e_k , substituting that into equation (A.6), performing the indicated expectations, taking the derivative of the trace of the result with respect to K , setting that result equal to zero, and then solving for K . For more details see [Maybeck79; Brown92; Jacobs93]. One form of the resulting K that minimizes equation (A.6) is given by¹

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ &= \frac{P_k^- H^T}{H P_k^- H^T + R} \end{aligned} \quad (\text{A.8})$$

Looking at equation (A.8) we see that as the measurement error covariance R approaches zero, the gain K weights the residual more heavily. Specifically,

$$\lim_{R_k \rightarrow 0} K_k = H^{-1}.$$

On the other hand, as the *a priori* estimate error covariance P_k^- approaches zero, the gain K weights the residual less heavily. Specifically,

$$\lim_{P_k^- \rightarrow 0} K_k = 0.$$

Another way of thinking about the weighting by K is that as the measurement error covariance R approaches zero, the actual measurement z_k is “trusted” more and more, while the predicted measurement $H\hat{x}_k$ is trusted less and less. On the other hand, as the *a priori* estimate error covariance P_k^- approaches zero the actual measurement z_k is trusted less and less, while the predicted measurement $H\hat{x}_k$ is trusted more and more.

A.1.3 The Probabilistic Origins of the Filter

The justification for equation (A.7) is rooted in the probability of the *a priori* estimate \hat{x}_k conditioned on all prior measurements z_k (Bayes’ rule). For now let it suffice to point out that the Kalman filter maintains the first two moments of the state distribution,

$$\begin{aligned} E[x_k] &= \hat{x}_k \\ E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] &= P_k. \end{aligned}$$

1. All of the Kalman filter equations can be algebraically manipulated into several forms.
Equation equation (A.8) represents the Kalman gain in one popular form.

The *a posteriori* state estimate equation (A.7) reflects the mean (the first moment) of the state distribution— it is normally distributed if the conditions of equation (A.3) and equation (A.4) are met. The *a posteriori* estimate error covariance equation (A.6) reflects the variance of the state distribution (the second non-central moment). In other words,

$$\begin{aligned} p(x_k | z_k) &\sim N(E[x_k], E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]) \\ &= N(\hat{x}_k, P_k). \end{aligned}$$

For more details on the probabilistic origins of the Kalman filter, see [Maybeck79; Brown92; Jacobs93].

A.1.4 The Discrete Kalman Filter Algorithm

We will begin this section with a broad overview, covering the “high-level” operation of one form of the discrete Kalman filter (see the previous footnote). After presenting this high-level view, we will narrow the focus to the specific equations and their use in this version of the filter.

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: *time update* equations and *measurement update* equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate.

The time update equations can also be thought of as *predictor* equations, while the measurement update equations can be thought of as *corrector* equations. Indeed the final estimation algorithm resembles that of a *predictor-corrector* algorithm for solving numerical problems as shown below in Figure A.1..

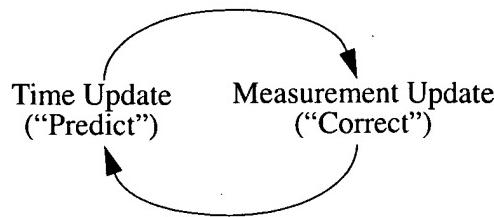


Figure A.1: The ongoing discrete Kalman filter cycle. The *time update* projects the current state estimate ahead in time. The *measurement update* adjusts the projected estimate by an actual measurement at that time.

The specific equations for the time and measurement updates are presented below in table A.1 and table A.2.

Table A.1: Discrete Kalman filter time update equations.

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_k \quad (\text{A.9})$$

$$\bar{P}_k = AP_{k-1}A^T + Q \quad (\text{A.10})$$

Again notice how the time update equations in table A.1 project the state and covariance estimates forward from time step $k - 1$ to step k . A and B are from equation (A.1), while Q is from equation (A.3). Initial conditions for the filter are discussed in the earlier references.

Table A.2: Discrete Kalman filter measurement update equations.

$$K_k = \bar{P}_k H^T (H \bar{P}_k H^T + R)^{-1} \quad (\text{A.11})$$

$$\hat{x}_k = \hat{x}_k + K_k (z_k - H \hat{x}_k) \quad (\text{A.12})$$

$$P_k = (I - K_k H) \bar{P}_k \quad (\text{A.13})$$

The first task during the measurement update is to compute the Kalman gain, K_k . Notice that the equation given here as equation (A.11) is the same as equation (A.8). The next step is to actually measure the process to obtain z_k , and then to generate an *a posteriori* state estimate by incorporating the measurement as in equation (A.12). Again equation (A.12) is simply equation (A.7) repeated here for completeness. The final step is to obtain an *a posteriori* error covariance estimate via equation (A.13).

After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This recursive nature is one of the very appealing features of the Kalman filter—it makes practical implementations much more feasible than (for example) an implementation of a Wiener filter [Brown92] which is designed to operate on *all* of the data *directly* for each estimate. The Kalman filter instead recursively conditions the current estimate on all of the past measurements. Figure A.2 below offers a complete picture of the operation of the filter, combining the high-level diagram of Figure A.1 with the equations from table A.1 and table A.2.

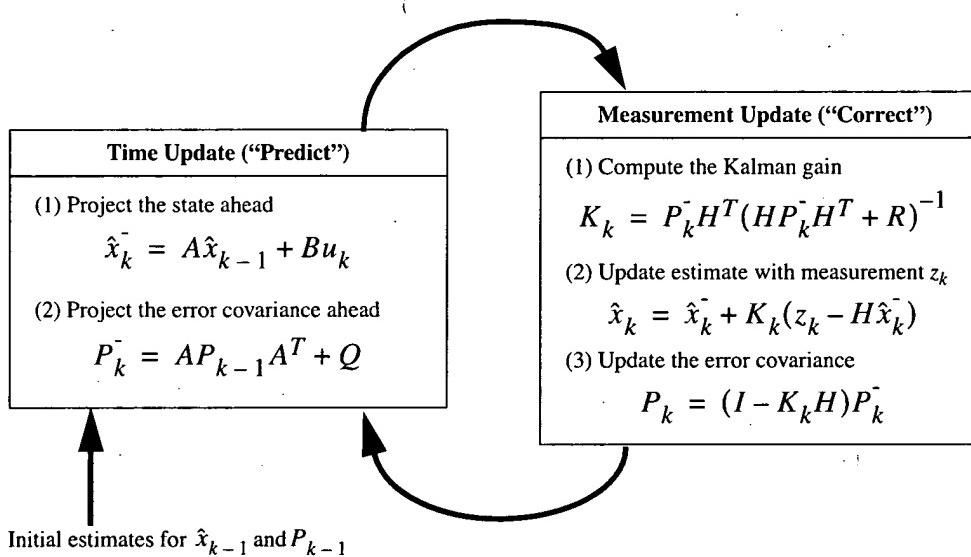


Figure A.2: A complete picture of the operation of the Kalman filter, combining the high-level diagram of Figure A.1 with the equations from table A.1 and table A.2.

A.2 The Extended Kalman Filter (EKF)

A.2.1 The Process to be Estimated

As described above in Section A.1.1, the Kalman filter addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by a *linear* stochastic difference equation. But what happens if the process to be estimated and (or) the measurement relationship to the process is non-linear? Some of the most interesting and successful applications of Kalman filtering have been such situations. A Kalman filter that linearizes about the current mean and covariance is referred to as an *extended Kalman filter* or EKF.

In something akin to a Taylor series, we can linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates even in the face of non-linear relationships. To do so, we must begin by

modifying some of the material presented in Section A.1. Let us assume that our process again has a state vector $x \in \Re^n$, but that the process is now governed by the *non-linear* stochastic difference equation

$$x_k = f(x_{k-1}, u_k, w_{k-1}), \quad (\text{A.14})$$

with a measurement $z \in \Re^m$ that is

$$z_k = h(x_k, v_k), \quad (\text{A.15})$$

where the random variables w_k and v_k again represent the process and measurement noise as in equation (A.3) and equation (A.4). In this case the *non-linear* function f in the difference equation equation (A.14) relates the state at the previous time step $k-1$ to the state at the current time step k . It includes as parameters any driving function u_k and the zero-mean process noise w_k . The *non-linear* function h in the measurement equation equation (A.15) relates the state x_k to the measurement z_k .

In practice of course one does not know the individual values of the noise w_k and v_k at each time step. However, one can approximate the state and measurement vector without them as

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_k, 0) \quad (\text{A.16})$$

and

$$\tilde{z}_k = h(\tilde{x}_k, 0), \quad (\text{A.17})$$

where \hat{x}_k is some *a posteriori* estimate of the state (from a previous time step k).

It is important to note that a fundamental flaw of the EKF is that the distributions (or densities in the continuous case) of the various random variables are no longer normal after undergoing their respective nonlinear transformations. The EKF is simply an *ad hoc* state estimator that only approximates the optimality of Bayes' rule by linearization. Some interesting work has been done by Julier et al. in developing a variation to the EKF, using methods that preserve the normal distributions throughout the non-linear transformations [Julier96].

A.2.2 The Computational Origins of the Filter

To estimate a process with non-linear difference and measurement relationships, we begin by writing new governing equations that linearize an estimate about equation (A.16) and equation (A.17),

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + Ww_{k-1}, \quad (\text{A.18})$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k. \quad (\text{A.19})$$

where

- x_k and z_k are the actual state and measurement vectors,
- \tilde{x}_k and \tilde{z}_k are the approximate state and measurement vectors from equation (A.16) and equation (A.17),
- \hat{x}_k is an *a posteriori* estimate of the state at step k ,
- the random variables w_k and v_k represent the process and measurement noise as in equation (A.3) and equation (A.4).
- A is the Jacobian matrix of partial derivatives of f with respect to x , that is

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_k, 0),$$

- W is the Jacobian matrix of partial derivatives of f with respect to w ,

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_k, 0),$$

- H is the Jacobian matrix of partial derivatives of h with respect to x ,

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0),$$

- V is the Jacobian matrix of partial derivatives of h with respect to v ,

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0).$$

Note that for simplicity in the notation we do not use the time step subscript k with the Jacobians A , W , H , and V , even though they are in fact different at each time step.

Now we define a new notation for the prediction error,

$$\tilde{e}_{x_k} \equiv x_k - \tilde{x}_k, \quad (\text{A.20})$$

and the measurement residual,

$$\tilde{e}_{z_k} \equiv z_k - \tilde{z}_k. \quad (\text{A.21})$$

Remember that in practice one does not have access to x_k in equation (A.20), it is the *actual* state vector, i.e. the quantity one is trying to estimate. On the other hand, one *does* have access to z_k in equation (A.21), it is the actual measurement that one is using to estimate x_k . Using equation (A.20) and equation (A.21) we can write governing equations for an *error process* as

$$\tilde{e}_{x_k} \approx A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k, \quad (\text{A.22})$$

$$\tilde{e}_{z_k} \approx H\tilde{e}_{x_k} + \eta_k, \quad (\text{A.23})$$

where ε_k and η_k represent new independent random variables having zero mean and covariance matrices WQW^T and VRV^T , with Q and R as in (A.3) and (A.4) respectively.

Notice that the equations equation (A.22) and equation (A.23) are linear, and that they closely resemble the difference and measurement equations equation (A.1) and equation (A.2) from the discrete Kalman filter. This motivates us to use the actual measurement residual \tilde{e}_{z_k} in equation (A.21) and a second (hypothetical) Kalman filter to estimate the prediction error \tilde{e}_{x_k} given by equation (A.22). This estimate, call it \hat{e}_k , could then be used along with equation (A.20) to obtain the *a posteriori* state estimates for the original non-linear process as

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k. \quad (\text{A.24})$$

The random variables of equation (A.22) and equation (A.23) have approximately the following probability distributions (see the previous footnote):

$$p(\tilde{e}_{x_k}) \sim N(0, E[\tilde{e}_{x_k} \tilde{e}_{x_k}^T])$$

$$p(\varepsilon_k) \sim N(0, WQ_k W^T)$$

$$p(\eta_k) \sim N(0, VR_k V^T)$$

Given these approximations and letting the predicted value of \hat{e}_k be zero, the Kalman filter equation used to estimate \hat{e}_k is

$$\hat{e}_k = K_k \tilde{e}_{z_k}. \quad (\text{A.25})$$

By substituting equation (A.25) back into equation (A.24) and making use of equation (A.21) we see that we do not actually need the second (hypothetical) Kalman filter:

$$\begin{aligned} \hat{x}_k &= \tilde{x}_k + K_k \tilde{e}_{z_k} \\ &= \tilde{x}_k + K_k(z_k - \tilde{z}_k) \end{aligned} \quad (\text{A.26})$$

Equation equation (A.26) can now be used for the measurement update in the extended Kalman filter, with \tilde{x}_k and \tilde{z}_k coming from equation (A.16) and equation (A.17), and the Kalman gain K_k coming from equation (A.11) with the appropriate substitution for the measurement error covariance.

The complete set of EKF equations is shown below in table A.3 and table A.4. Note that we have substituted \tilde{x}_k for \hat{x}_k to remain consistent with the earlier “super minus” a priori notation, and that we now attach the subscript k to the Jacobians A , W , H , and V , to reinforce the notion that they are different at (and therefore must be recomputed at) each time step.

Table A.3: EKF time update equations.

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_k, 0) \quad (\text{A.27})$$

$$\tilde{P}_k = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \quad (\text{A.28})$$

As with the basic discrete Kalman filter, the time update equations in table A.3 project the state and covariance estimates from the previous time step $k - 1$ to the current time step k . Again f in equation (A.27) comes from equation (A.16), A_k and W_k are the process Jacobians at step k , and Q_k is the process noise covariance equation (A.3) at step k .

Table A.4: EKF measurement update equations.

$$K_k = \tilde{P}_k H_k^T (H_k \tilde{P}_k H_k^T + V_k R_k V_k^T)^{-1} \quad (\text{A.29})$$

$$\hat{x}_k = \tilde{x}_k + K_k (z_k - h(\tilde{x}_k, 0)) \quad (\text{A.30})$$

$$P_k = (I - K_k H_k) \tilde{P}_k \quad (\text{A.31})$$

As with the basic discrete Kalman filter, the measurement update equations in table A.4 correct the state and covariance estimates with the measurement z_k . Again h in equation (A.30) comes from equation (A.17), H_k and V are the measurement Jacobians at step k , and R_k is the measurement noise covariance equation (A.4) at step k . (Note we now subscript R allowing it to change with each measurement.)

The basic operation of the EKF is the same as the linear discrete Kalman filter as shown in Figure A.1. Figure A.3 below offers a complete picture of the operation of the EKF, combining the high-level diagram of Figure A.1 with the equations from table A.3 and table A.4.

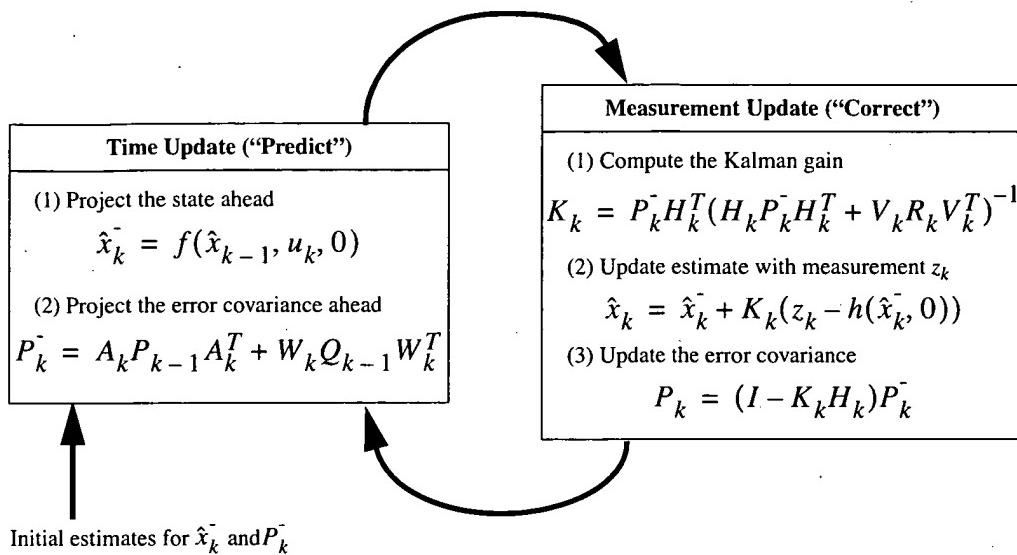


Figure A.3: A complete picture of the operation of the *extended* Kalman filter, combining the high-level diagram of Figure A.1 with the equations from table A.3 and table A.4.

An important feature of the EKF is that the Jacobian H_k in the equation for the Kalman gain K_k serves to correctly propagate or “magnify” only the relevant component of the measurement information. For example, if there is not a one-to-one mapping between the measurement z_k and the state via h , the Jacobian H_k affects the Kalman gain so that it only magnifies the portion of the residual $z_k - h(\hat{x}_k, 0)$ that does affect the state. Of course if over *all* measurements there is *not* a one-to-one mapping between the measurement z_k and the state via h , then as you might expect the filter will quickly diverge. In this case the process is *unobservable*.

A.3 An Example: Estimating a Random Constant

In the previous two sections we presented the basic form for the discrete Kalman filter, and the extended Kalman filter. To help in developing a better feel for the operation and capability of the filter, we present a very simple example here.

A.3.1 The Process Model

In this simple example let us attempt to estimate a scalar random constant, a voltage for example. Let's assume that we have the ability to take measurements of the constant, but that the measurements are corrupted by a 0.1 volt RMS *white* measurement noise (e.g. our analog to digital converter is not very accurate). In this example, our process is governed by the linear difference equation

$$\begin{aligned}x_k &= Ax_{k-1} + Bu_k + w_k \\&= x_{k-1} + w_k\end{aligned}$$

with a measurement $z \in \mathbb{R}^1$ that is

$$\begin{aligned}z_k &= Hx_k + v_k \\&= x_k + v_k\end{aligned}$$

The state does not change from step to step so $A = 1$. There is no control input so $u = 0$. Our noisy measurement is of the state directly so $H = 1$. (Notice that we dropped the subscript k in several places because the respective parameters remain constant in our simple model.)

A.3.2 The Filter Equations and Parameters

Our time update equations are

$$\hat{x}_k^- = \hat{x}_{k-1},$$

$$P_k^- = P_{k-1} + Q,$$

and our measurement update equations are

$$\begin{aligned}K_k &= P_k^- (P_k^- + R)^{-1} \\&= \frac{P_k^-}{P_k^- + R}\end{aligned} \tag{A.32}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - \hat{x}_k^-),$$

$$P_k = (1 - K_k) P_k^-.$$

Presuming a very small process variance, we let $Q = 1e-5$. (We could certainly let $Q = 0$ but assuming a small but non-zero value gives us more flexibility in “tuning” the filter as we will demonstrate below.) Let's assume that from experience we know that the

true value of the random constant has a standard normal probability distribution, so we will “seed” our filter with the guess that the constant is 0. In other words, before starting we let $\hat{x}_{k-1} = 0$.

Similarly we need to choose an initial value for P_{k-1} , call it P_0 . If we were absolutely certain that our initial state estimate $\hat{x}_0 = 0$ was correct, we would let $P_0 = 0$. However given the uncertainty in our initial estimate \hat{x}_0 , choosing $P_0 = 0$ would cause the filter to initially and always believe $\hat{x}_k = 0$. As it turns out, the alternative choice is not critical. We could choose almost any $P_0 \neq 0$ and the filter would *eventually* converge. We’ll start our filter with $P_0 = 1$.

A.3.3 The Simulations

To begin with, we randomly chose a scalar constant $z = -0.37727$ (there is no “hat” on the z because it represents the “truth”). We then simulated 50 distinct measurements z_k that had error normally distributed around zero with a standard deviation of 0.1 (remember we presumed that the measurements are corrupted by a 0.1 volt RMS *white* measurement noise). We could have generated the individual measurements within the filter loop, but pre-generating the set of 50 measurements allowed me to run several simulations with the same exact measurements (i.e. same measurement noise) so that comparisons between simulations with different parameters would be more meaningful.

In the first simulation we fixed the measurement variance at $R = (0.1)^2 = 0.01$. Because this is the “true” measurement error variance, we would expect the “best” performance in terms of balancing responsiveness and estimate variance. This will become more evident in the second and third simulation. Figure A.4 depicts the results of this first simulation. The true value of the random constant $x = -0.37727$ is given by the solid line, the noisy measurements by the cross marks, and the filter estimate by the remaining curve.

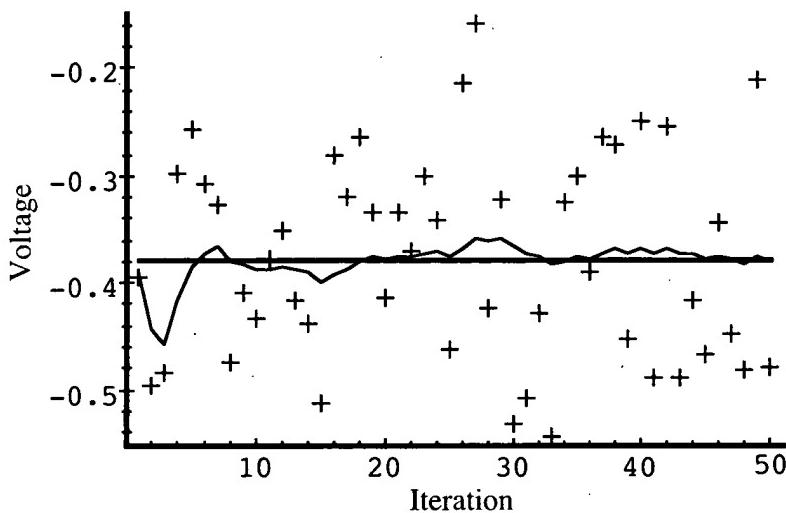


Figure A.4: The first simulation: $R = (0.1)^2 = 0.01$. The true value of the random constant $x = -0.37727$ is given by the solid line, the noisy measurements by the cross marks, and the filter estimate by the remaining curve.

When considering the choice for P_0 above, we mentioned that the choice was not critical as long as $P_0 \neq 0$ because the filter would eventually converge. Below in Figure A.5 we have plotted the value of P_k versus the iteration. By the 50th iteration, it has settled from the initial (rough) choice of 1 to approximately 0.0002 (Volts²).

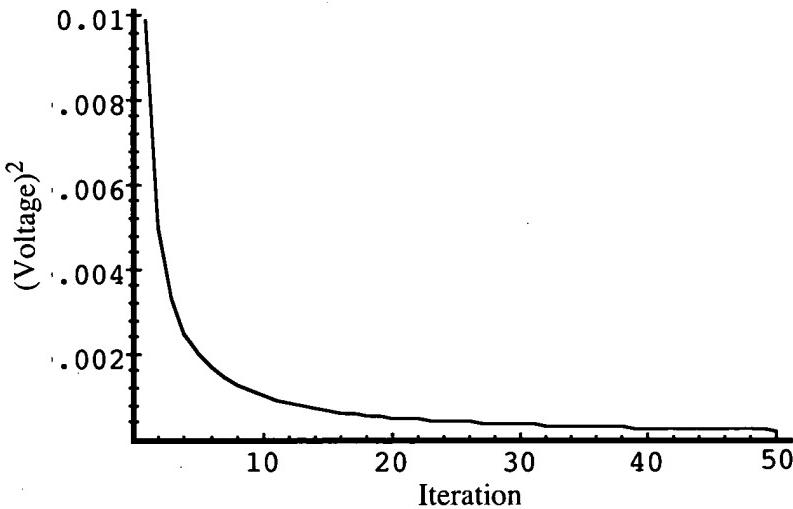


Figure A.5: After 50 iterations, our initial (rough) error covariance P_k choice of 1 has settled to about 0.0002 (Volts²).

In Figure A.6 and Figure A.7 below we can see what happens when R is increased or decreased by a factor of 100 respectively. In Figure A.6 the filter was told that the measurement variance was 100 times greater (i.e. $R = 1$) so it was “slower” to believe the measurements.

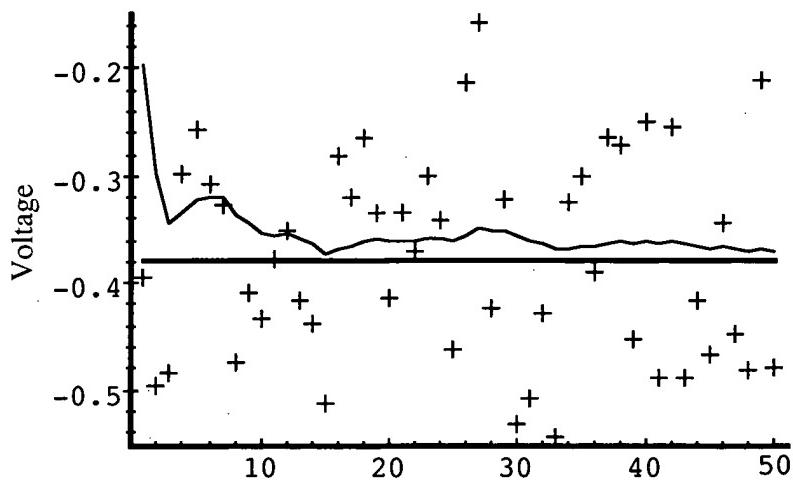


Figure A.6: Second simulation: $R = 1$. The filter is slower to respond to the measurements, resulting in reduced estimate variance.

In Figure A.7 the filter was told that the measurement variance was 100 times smaller (i.e. $R = 0.0001$) so it was very “quick” to believe the noisy measurements.

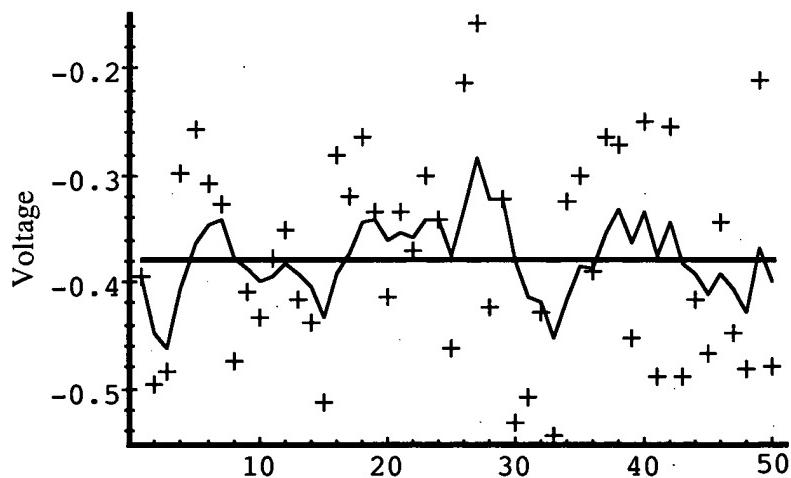


Figure A.7: Third simulation: $R = 0.0001$. The filter responds to measurements quickly, increasing the estimate variance.

While the estimation of a constant is relatively straight-forward, it clearly demonstrates the workings of the Kalman filter. In Figure A.6 in particular the Kalman “filtering” is evident as the estimate appears considerably smoother than the noisy measurements.

References

- Brown92 Brown, R. G. and P. Y. C. Hwang. 1992. *Introduction to Random Signals and Applied Kalman Filtering, Second Edition*, John Wiley & Sons, Inc.
- Gelb74 Gelb, A. 1974. *Applied Optimal Estimation*, MIT Press, Cambridge, MA.
- Grewal93 Grewal, Mohinder S., and Angus P. Andrews (1993). Kalman Filtering Theory and Practice. Upper Saddle River, NJ USA, Prentice Hall.
- Jacobs93 Jacobs, O. L. R. 1993. *Introduction to Control Theory, 2nd Edition*. Oxford University Press.
- Julier96 Julier, Simon and Jeffrey Uhlman. "A General Method of Approximating Nonlinear Transformations of Probability Distributions," Robotics Research Group, Department of Engineering Science, University of Oxford [cited 14 November 1995]. Available from <http://www.robots.ox.ac.uk/~siju/work/publications/Unscented.zip>.
Also see: "A New Approach for Filtering Nonlinear Systems" by S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, Proceedings of the 1995 American Control Conference, Seattle, Washington, Pages:1628-1632. Available from http://www.robots.ox.ac.uk/~siju/work/publications/ACC95_pr.zip.
- Also see Simon Julier's home page at <http://www.robots.ox.ac.uk/~siju/>.
- Kalman60 Kalman, R. E. 1960. "A New Approach to Linear Filtering and Prediction Problems," Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960).
- Lewis86 Lewis, Richard. 1986. *Optimal Estimation with an Introduction to Stochastic Control Theory*, John Wiley & Sons, Inc.
- Maybeck79 Maybeck, Peter S. 1979. *Stochastic Models, Estimation, and Control, Volume 1*, Academic Press, Inc.
- Sorenson70 Sorenson, H. W. 1970. "Least-Squares estimation: from Gauss to Kalman," *IEEE Spectrum*, vol. 7, pp. 63-68, July 1970.

B. Tracking Bibliography

This appendix includes a listing of most of the tracking-related publications that we are aware of and refer to. This list is also available in electronic bibliography format at

<http://www.cs.unc.edu/~tracker/ref/s2001/tracker/>

-
- 3rdTech. (2000, July 15). *3rdTech™*, [HTML]. 3rdTech. Available: <http://www.3rdtech.com/> [2000, July 19].
- Agar, W. O., & Blythe, J. H. (1968). An optical method of measuring transverse surface velocity. *Journal of Scientific Instruments (Journal of Physics E) 1968 Series 2, 1*, 25-28.
- Aidala, V. J. (1979). Kalman filter behavior in bearings-only tracking applications. *IEEE Transactions on Aerospace and Electronic Systems, AES-15*(1), 29-39.
- Aidala, V. J., & Hammel, S. E. (1983). Utilization of modified polar coordinates for bearings-only tracking. *IEEE Trans. Automat. Contr., AC-28*, 283-294.
- Akatsuka, Y., & Bekey, G. A. (1998). Compensation for end to end delays in a VR system, *Proceedings of IEEE VRAIS'98* (pp. 156-159). Atlanta, GA: IEEE.
- Antonsson, E. K., & Mann, R. W. (1989). Automatic 6-D.O.F. kinematic trajectory acquisition and analysis. *Journal of Dynamic Systems, Measurement, and Control, 111*, 31-39.
- Ascension. (2000). *Ascension Technology Corporation*, [HTML]. Ascension Technology Corporation. Available: <http://www.ascension-tech.com/> [2000, September 15].
- Atkeson, C. G., & Hollerbach, J. M. (1985). Kinematic features of unrestrained vertical arm movements. *Journal of Neuroscience, 5*, 2318-2330.
- Ator, J. T. (1963). Image-velocity sensing with parallel-slit recticles. *Journal of the Optical Society of America, 53*(12), 1416-1422.
- Ator, J. T. (1966). Image velocity sensing by optical correlation. *Applied Optics, 5*(8), 1325-1331.
- Azarbajayani, A., & Pentland, A. (1994). *Recursive estimation of motion, structure, and focal length* (Technical report 243). Cambridge, MA: Massachusetts Institute of Technology (MIT).
- Azarbajayani, A., & Pentland, A. (1995a). *Camera self-calibration from one point correspondence* (Perceptual Computing Technical Report 341): MIT Media Laboratory.

- Azarbajiani, A., & Pentland, A. (1995b). Recursive Estimation of Motion, Structure, and Focal Length. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(6), 562-575.
- Azarbajiani, A., & Pentland, A. (1996). *Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features* (Technical report 363). Cambridge, MA: Massachusetts Institute of Technology (MIT).
- Azuma, R. T. (1993, July). Tracking Requirements for Augmented Reality. *Communications of the ACM*, 36, 50-51.
- Azuma, R. T. (1995a). *Predictive Tracking for Augmented Reality*. Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC USA.
- Azuma, R. T. (1995b). *Predictive Tracking for Augmented Reality* (TR95-007). Chapel Hill, NC: University of North Carolina at Chapel Hill, Department of Computer Science.
- Azuma, R. T. (1995c). *A survey of augmented reality*. Unpublished manuscript, Malibu, CA.
- Azuma, R. T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 355-385.
- Azuma, R. T. (1999). The Challenge of Making Augmented Reality Work Outdoors. In Y. Ohta & H. Tamura (Eds.), *Mixed Reality: Merging Real and Virtual Worlds* (pp. 379-390). Yokohama, Japan: Springer-Verlag.
- Azuma, R. T., & Bishop, G. (1994). Improving Static and Dynamic Registration in an Optical See-Through HMD, *Computer Graphics* (SIGGRAPH 94 Conference Proceedings ed., pp. 197-204). Orlando, FL USA: ACM Press, Addison-Wesley.
- Azuma, R. T., & Bishop, G. (1995). A Frequency-Domain Analysis of Head-Motion Prediction, *Computer Graphics* (SIGGRAPH 94 Conference Proceedings ed., pp. 401-408). Los Angeles, CA: ACM Press, Addison-Wesley.
- Azuma, R. T., Hoff, B. R., Neely, H. E., III, Sarfaty, R., Daily, M. J., Bishop, G., Chi, V., Welch, G., Neumann, U., You, S., Nichols, R., & Cannon, J. (1998). Making Augmented Reality Work Outdoors Requires Hybrid Tracking, *First International Workshop on Augmented Reality* (pp. 219-224). San Francisco, CA, USA.
- Azuma, R. T., Hoff, B. R., & Neely, H. E. I. (1999). A Motion-Stabilized Outdoor Augmented Reality System, *IEEE Virtual Reality* (pp. 252-259). Houston, TX USA.
- Azuma, R. T., Lee, J. W., Jiang, B., Park, J., You, S., & Neumann, U. (1999). Tracking in unprepared environments for augmented reality systems. *Computers & Graphics*, 23(6), 787-793.
- Azuma, R. T., & Ward, M. (1991). *Space-Resection by Collinearity: Mathematics Behind the Optical Ceiling Head-Tracker* (Technical Report 91-048). Chapel Hill, NC USA: University of North Carolina at Chapel Hill.
- Bachmann, E., Robert. (2000). *Inertial and Magnetic Tracking of Limb Segment Orientation for Inserting Humans into Synthetic Environments*. Unpublished Ph.D. Thesis, The Naval Postgraduate School, Monterey, CA.

- Bachmann, E., Robert, Duman, I., McGhee, R. B., & Zyda, M. J. (1999). *Sourceless Sensing of Limb Segment Angles for Inserting Humans Into Networked Virtual Environment*: The Naval Postgraduate School.
- Bailey, T., Nebot, E., M., Rosenblatt, J. K., & Durrant-Whyte, H., F. (1999). Robust distinctive place recognition for topological maps, *International Conference on Field and Service Robotics (FSR 99)*. Pittsburgh, PA USA.
- Bajura, M., & Neumann, U. (1995). Closed-Loop Tracking for Augmented-Reality Systems. *IEEE Computer Graphics & Applications*, 15(5), 52-60.
- Bajura, M., & Neumann, U. (2001). *Dynamic Compensation of Alignment Error in Augmented-Reality Systems*, [HTML]. Available: http://www.usc.edu/dept/CGIT/papers/VRpose_94_022.pdf [2001, March 24, 2001].
- Bancroft, S. (1984). An algebraic solution of the GPS equations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(7), 56-59.
- Bar-Shalom, Y., & Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software*: Artec House, Inc.
- Baron, S., Lancraft, R., & Caglayan, A. (1984). *An optimal control model approach to the design of compensators for simulator delay* (NASA Contractor Report 3064). Cambridge, MA: Bolt Beranek and Newman Inc.
- Behringer, R. (1999). Registration for Outdoor Augmented Reality Applications Using Computer Vision Techniques and Hybrid Sensors, *IEEE Virtual Reality* (pp. 244-251). Houston, TX, USA.
- Bell, B. M., & Cathey, F. W. (1993). The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2), 294-297.
- Best, R. E. (1999). Phase-Locked Loops: Design, Simulation, and Applications.
- Bhatnagar, D. K. (1993). *Position trackers for Head Mounted Display systems: A survey* (Technical Report TR93-010). Chapel Hill, NC USA: University of North Carolina at Chapel Hill.
- Bible, S. R., Zyda, M., & Brutzman, D. Using spread-spectrum ranging techniques for position tracking in a virtual environment (pp. 15).
- Bishop, G. (1984). *The Self-Tracker: A Smart Optical Sensor on Silicon*. Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC USA.
- Bishop, G., & Fuchs, H. *Self-Tracker: a VLSI-based three-dimensional input system* (paper 83-002). Chapel Hill, NC: University of North Carolina at Chapel Hill Department of Computer Science.
- Bishop, G., & Fuchs, H. (1984). The Self-Tracker: A Smart Optical Sensor on Silicon, *Advanced Research in VLSI* (pp. 65-73). Massachusetts Institute of Technology: Artech House.
- BL. (2000). *CODA mpx30 Motion Capture System*, [html]. B & L Engineering. Available: <http://www.charndyn.com/> and <http://www.bleng.com/coda.htm> [2000, April 27].
- Bolles, R. C., Kremers, J. H., & Cain, R. A. (1981). *A simple sensor to gather three-dimensional data* (Technical Note 249)

- SRI Project 1538). Menlo Park, CA: SRI International, Industrial Automation Department, Computer Science and Technology Division.
- Bordtad, A. J. (1985). Bearings-only target motion analysis estimation characteristics. *Control and Computers*, 13(3), 95-101.
- Borghese, N. A., & Ferrigno, G. (1990). An algorithm for 3-D automatic movement detection by means of standard TV cameras. *IEEE Transactions on Biomedical Engineering*, 37(12), 1221-1225.
- Bouget, J.-Y. (1997). 3D transformations & camera calibration (pp. 7 pp).
- Britannica, E. (1994). *Encyclopedia Britannica*, [HTML]. Encyclopedia Britannica. Available: <http://www.britannica.com/> [2001, April 27].
- Broida, T. J., & Chellappa, R. (1986). Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1), 90-99.
- Brown, R. G., & Hwang, P. Y. C. (1992). *Introduction to Random Signals and Applied Kalman Filtering* (Second ed.): Wiley & Sons, Inc.
- Brown, R. G., & Hwang, P. Y. C. (1996). *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises and Solutions* (Third ed.): Wiley & Sons, Inc.
- Brown, R. G., & Hwang, P. Y. C. (1997). Introduction to Random Signals and Applied {K}alman Filtering: with {MATLAB} Exercises and Solutions.
- Brugger, W., & Milner, M. (1978). Computer-aided tracking of body motions using a c.c.d.-image sensor. *Medical & Biological Engineering & Computing*, 16, 207-210.
- Bruss, A. R., & Horn, B. K. P. (1981). *Passive navigation* (A.I. Memo 662). Cambridge, MA: Massachusetts Institute of Technology Artificial Intelligence Laboratory.
- Bryson, S. (1992). SPIE Proceedings Volume 1669 Stereoscopic Displays and Applications III. In J. O. Merritt & S. S. Fisher (Eds.), *SPIE* (Vol. 1669, pp. 244-255). San Jose, CA.
- Bui, H. H., Vankatesh, S., & West, G. (2000). A probabilistic framework for tracking in wide-area environments, *Proceedings of the International Conference on Pattern Recognition (ICPR'00)* (Vol. 4, pp. 702-706). Barcelona, Spain.
- Burdea, G., & Coiffet, P. (1994). *Virtual Reality Technology* (First ed.): John Wiley & Sons, Inc.
- Burton, R. P. (1973). *Real-Time Measurement of Multiple Three-Dimensional Positions*. Unpublished Ph.D., University of Utah, Salt Lake City, UT USA.
- Burton, R. P., & Sutherland, I. E. (1974). TWINKLEBOX: A Three-Dimensional Computer-Input Device, *AFIPS Conference Proceedings, 1974 National Computer Conference* (Vol. 43, pp. 513-520). Chicago, IL USA: AFIPS Press, Montvale, New Jersey.
- Card, S. K., Mackinlay, J. D., & Robertson, G. G. (1991). A Morphological Analysis of the Design Space of Input Devices. 9(2), 99-122.
- Cavallaro, R. (1997). The FoxTrax hockey puck tracking system, *IEEE CG&A* (pp. 6-12).

- Chi, V. L. (1995). *Noise Model and Performance Analysis Of Outward-looking Optical Trackers Using Lateral Effect Photo Diodes* (TR95-012). Chapel Hill, NC USA: University of North Carolina at Chapel Hill.
- Chou, J. C. K. (1992). Quaternion kinematic and dynamic equations. *IEEE Transactions on Robotics and Automation*, 8(1), 53-64.
- Corporation, A. (2001, 2000). *Gypsy Motion Capture System*, [HTML]. Analogus Corporation. Available: <http://www.metamotion.com/gypsy-motion-capture-system/gypsy-motion-capture-system.htm> [2001, March 7].
- Council, N. R. (1994). *Virtual Reality, Scientific and Technological Challenges*. Washington, DC: National Academy Press.
- Crane, D. F. (1980). The effects of time delay in man-machine control systems: Implementations for design of flight simulator-display-delay compensation, *IMAGE III* (pp. 331-343). Williams AFB, AZ: Air Force Human Resources Laboratory.
- Crowley, J. L., & Demazeau, Y. (1993). Principles and Techniques for Sensor Data Fusion. *Signal Processing (EURASIP)*, 32, 5-27.
- Darrell, T., Azarbayejani, A., & Pentland, A. P. (1994). *Segmentation of rigidly moving objects using multiple Kalman filters* (Technical report 281). Cambridge, MA: Massachusetts Institute of Technology (MIT).
- David, P., Balakirsky, S., & Hillis, D. (1990). A real-time automatic target acquisition system, *Symposium of the Association for Unmanned Vehicle Systems* (pp. 13 pp). Dayton, OH.
- De Geeter, J., Van Brussel, H., De Schutter, J., & Decreton, M. (1996). Recognising and locating objects with local sensors, *IEEE Conference on Robotics and Automation* (pp. 6 pp (In conf. proceedings: 3478-3483)). Minneapolis, Minnesota.
- Deering, M. F. (1992). High resolution virtual reality. In E. E. Catmull (Ed.), *Computer Graphics (Proceedings of SIGGRAPH 92)* (Vol. 26, pp. 195-202). Chicago, Illinois.
- Deyst, J., J., & Price, C. F. (1968). Conditions for Asymptotic Stability of the Discrete Minimum-Variance Linear Estimator. *IEEE Transactions on Automatic Control*.
- Division, S. D. I. (2001). *Gyrochip theory of operation*, [HTML]. BEI Electronics Company, Systron Donner Inertial Division [2001, April 27].
- Donner, S. (2001). *Systron Donner Home Page*, [HTML]. Available: <http://www.systron.com/> [2001, April 27].
- Dowski, E., R. (1995). An Information Theory Approach to Incoherent Information Processing Systems. *Signal Recovery and Synthesis V, OSA Technical Digest Series*, 106-108.
- Drane, C., R. (1992). *Positioning Systems : A Unified Approach* (Vol. 181): Springer Verlag.
- Durlach, N., & Mavor, A. S. (1994). Position tracking and mapping, *Virtual reality: scientific and technological challenges* (pp. 188-204). Washington, D.C.: National Academy Press.

- Ellis, S. R., Adelstein, B. D., Baumeler, S., Jense, G. J., & Jacoby, R. H. (1999). Sensor spatial distortion, visual latency, and update rate effects on 3D tracking in virtual environments, *Proceedings of the IEEE Virtual Reality*. Houston, Texas: Institute of Electrical and Electronics Engineers.
- Ellis, S. R., & Menges, B. M. (1997). Judgements of the distance to nearby virtual objects: interaction of viewing conditions and accommodative demand. *Presence: Teleoperators and Virtual Environments*, 6(4), 452-460.
- Emura, S., & Tachi, S. (1994a). Compensation of time lag between actual and virtual spaces by multi-sensor integration, *1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (pp. 363-469). Las Vegas, NV: Institute of Electrical and Electronics Engineers.
- Emura, S., & Tachi, S. (1994b). *Sensor Fusion based Measurement of Human Head Motion*. Paper presented at the 3rd IEEE International Workshop on Robot and Human Communication (RO-MAN 94 NAGOYA), Nagoya University, Nagoya, Japan.
- Emura, S., & Tachi, S. (1994). Sensor Fusion based Measurement of Human Head Motion, *3rd IEEE International Workshop on Robot and Human Communication (RO-MAN 94 NAGOYA)* (pp. 124-129). Nagoya University, Nagoya, Japan.
- Etienne-Cummings, R., Spiegel, V. d., & Mueller, P. (1997). A focal plane visual motion measurement sensor. *IEEE Transactions on Circuits and Systems*, 44(1), 55-66.
- Falconer, D. G. (1979). *Target tracking with the Hough and Fourier-Hough transform* (Technical Note 202). Menlo Park, CA: SRI International.
- Faugeras, O. (1999). *Three-dimensional computer vision: a geometric viewpoint* (3rd ed. Vol. 1). Cambridge, Massachusetts: The MIT Press.
- Feder, H. J. S., Leonard, J. J., & Smith, C. M. (1999). Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18(7), 650-668.
- Feiner, S., MacIntyre, B., & Höllerer, T. (1997). A touring machine: prototyping 3D mobile augmented reality systems for exploring the urban environment, *Proceedings of First International Symposium on Wearable Computers* (pp. 74-81). Cambridge, MA.
- Feiner, S., MacIntyre, B., Höllerer, T., & Webster, A. (1997). A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring Urban Environments. *Personal Technologies*, 1(4), 208-217.
- Ferrin, F. J. (1991). Survey of Helmet Tracking Technologies, *Large-Screen-Projection, Avionic, and Helmet-Mounted Displays* (Vol. 1456, pp. 86--94).
- Fischer, P., Daniel, R., & Siva, K. (1990). Specification and Design of Input Devices for Teleoperation, *Proceedings of the IEEE Conference on Robotics and Automation* (pp. 540-545). Cincinnati, OH.
- Fischler, M. A., & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6), 381-395.
- Fleming, R., & Kushner, C. (1995). Low-Power, Miniature, Distributed Position Location and Communication Devices Using Ultra-Wideband, Nonsinusoidal Communication Technology.

- Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1997). *Computer Graphics: Principles and Practice* (2nd ed. Vol. 1). Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.
- Fox, D., Burgard, W., & Thrun, S. (1999). Probabilistic Methods for Mobile Robot Mapping, *IJCAI-99 Workshop on Adaptive Spatial Representations of Dynamic Environments* (pp. 10).
- Foxlin, E. (1993). *Inertial Head-Tracking*. Unpublished Master of Science, Massachusetts Institute of Technology (MIT), Cambridge, MA.
- Foxlin, E. (1996). Intertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter, *VRAIS 96* (pp. 185-194). Los Alamitos, CA USA: IEEE.
- Foxlin, E., & Durlach, N. (1994). An Inertial Head-Orientation Tracker with Automatic Drift Compensation for Use With HMD's. In G. Singh & S. Feiner & D. Thalmann (Eds.), *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (pp. 159-173). Singapore: ACM SIGGRAPH, Addison-Wesley.
- Foxlin, E., Harrington, M., & Altshuler, Y. (1998). Miniature 6-DOF Inertial System for Tracking HMDs, *SPIE Helmet and Head-Mounted Displays III* (Vol. 3362). Orlando, FL USA: SPIE.
- Foxlin, E., Harrington, M., & Pfeifer, G. (1998). Constellation™ : A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. In M. F. Cohen (Ed.), *Computer Graphics* (SIGGRAPH 98 Conference Proceedings ed., pp. 371-378). Orlando, FL USA: ACM Press, Addison-Wesley.
- Frey, W., Zyda, M., McGhee, R., & Cockayne, W. (1996). *Off-the-Shelf, Real-Time, Human Body Motion Capture for Synthetic Environments* (Technical Report NPSCS-96-003). Monterey, CA, USA: The Naval Postgraduate School.
- Friedman, M., Starner, T., & Pentland, A. (1992). Synchronization in Virtual Realities. *Presence: Teleoperators and Virtual Environments*, 1, 139-144.
- Friedmann, M., Starner, T., & Pentland, A. (1992). Device Synchronization Using an Optimal Linear Filter, *1992 Symposium on Interactive 3D Graphics*. Cambridge, MA USA.
- Frye, W. E. (1957 ?). Fundamentals of inertial guidance and navigation. *Journal of the Astronautical Sciences*, 1-10.
- Fuchs (Foxlin), E. (1993). *Inertial Head-Tracking*. Unpublished M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA USA.
- Fuchs, H., Duran, J., & Johnson, B. (1977). A system for automatic acquisition of three-dimensional data, *AFIPS Conf. Proc.* (Vol. 46, pp. 49-53).
- Fuhrmann, A., Loffelmann, H., & Schmalstieg, D. (1997). Collaborative augmented reality: exploring dynamical systems, *Proceedings of IEEE Visualization '97* (pp. 459-462). Phoenix, AZ.
- Ganapathy, S. Decomposition of transformation matrices for robot vision (pp. 21).
- Ganapathy, S. (1984). *Real-time motion tracking using a single camera* (Technical memorandum 11358-841105-21-TM, Charge case 311306-0399, File case 39394): AT&T Bell Laboratories.

- Ganapathy, S. (1994). *Camera Location Determination Problem* (Technical Memorandum 11358-841102-20-TM): AT&T Bell Laboratories Technical Memorandum.
- Gelb, A. (1974). *Applied Optimal Estimation*. Cambridge, MA: MIT Press.
- Gillis, J. T. (1991). Examination of 3-D angular motion using gyroscopes and linear accelerometers. *IEEE Transactions on Aerospace and Electronic Systems*, 27(6), 910-920.
- Golding, A. R., & Lesh, N. (1999). *Indoor navigation using a diverse set of cheap, wearable sensors* (Technical Report TR99-32). Cambridge, MA, USA: Mitsubishi Electric Information Technology Center America.
- Goncalves, L., Di Bernardo, E., Ursella, E., & Perona, P. (1995). Monocular tracking of the human arm in 3D, *ICCV'95* (pp. 7 pp (In conference proceedings: 764-770)).
- Goshtasby, A. A. (2001). *Active Vision*. Available: <http://www.cs.wright.edu/people/faculty/agoshtas/3.activevis/ActiveVis.html> [2001, April 27].
- Gottschalk, S., & Hughes, J. F. (1993). Autocalibration for Virtual Environments Tracking Hardware. In J. T. Kajiya (Ed.), *Computer Graphics* (SIGGRAPH 93 Conference Proceedings ed., pp. 65-72). Anaheim, CA USA: ACM Press, Addison Wesley.
- Grewal, M., S., & Andrews, A., P. (1993). *Kalman Filtering Theory and Practice*. Upper Saddle River, NJ USA: Prentice Hall.
- Grewal, M., S., & Andrews, A., P. (2001). *Kalman Filtering Theory and Practice Using MATLAB* (Second ed.). New York, NY USA: John Wiley & Sons, Inc.
- Grewal, M. S., Weill, L., R., & Andrews, A. P. (2001). *Global Positioning Systems, Inertial Navigation, and Integration*. New York, NY USA: John Wiley & Sons, Inc.
- Guivant, J., Nebot, E., M., & Durrant-Whyte, H., F. (2000). Simultaneous localization and map building using natural features in outdoor environments, *IAS-6 Intelligent Autonomous Systems*. Italy.
- Ham, F. M., & Brown, R. G. (1983). Observability, eigenvalues, and Kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19(2), 269-273.
- Hamilton, W. R. (1853). *Lectures on Quaternions*. Dublin: Hodges and Smith.
- Held, R., & Durlach, N. (1991). Telepresence, time delay, and adaptation. In S. R. Ellis (Ed.), *Pictorial Communication in Virtual and Real Environments* (pp. 28-21 - 28-16): Taylor and Francis.
- Herring, T. A. (1996). The global positioning system. *Scientific American*, 44-50.
- Hill, P. D., & Walsh, T. R. (1992). The advanced modular tracker: a real-time video tracker. Danvers, MA: Datacube, Inc.
- Hoff, W. A., & Nguyen, K. (1996). Computer vision-based registration techniques for augmented reality, *Proceedings of Intelligent Robots and Computer Vision XV, SPIE* (Vol. 2904, pp. 538-548). Boston, MA.
- Hohl, F., Kubach, U., Leonhardi, A., Rothermel, K., & Schwehm, M. (1999a). *Next century challenges: Nexus--an open global infrastructure for spatial-aware applications*. Paper presented at the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking.

- Hohl, F., Kubach, U., Leonhardi, A., Rothermel, K., & Schwehm, M. (1999b). Next century challenges: Nexus--an open global infrastructure for spatial-aware applications, *Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking* (pp. 249–255).
- Hohl, F., Kubach, U., Leonhardi, A., Rothermel, K., & Schwehm, M. (1999c). Nexus - An open global infrastructure for spatial-aware applications (pp. 13 pp).
- Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., & Hallaway, D. (1999). Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics*, 23(6), 779-785.
- Holloway, R. (1997). Registration error analysis for augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 413-432.
- Holloway, R. L. (1995). *Registration Errors in Augmented Reality Systems*. Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.
- Huang, T. S., & Netravali, A. N. (1994). Motion and structure from feature correspondences: a review, *Proceedings of the IEEE* (Vol. 82, pp. 252-267).
- Ickes, B. P. (1970). A new method for performing digital control systems attitude computations using quaternions. *AIAA Journal*, 5(1), 13-17.
- IGT. (2000). *FlashPoint 5000*, [HTML]. Image Guided Technologies. Available: <http://www.imageguided.com/> [2000, September 15].
- Iltanen, M., Kosola, H., Palovuori, K., & Vanhala, J. (1998). Optical Positioning and Tracking System for a Head Mounted Display Based on Spread Spectrum Technology, *Proceedings of the 2nd International Conference on Machine Automation (ICMA)* (pp. 597--608).
- Inigo, R. M., & McVey, E. S. (1981). CCD Implementation of a three-dimensional video-tracking algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-3*(2), 230-240.
- Intersense. (2000). *Intersense IS-900*, [html]. Intersense. Available: <http://www.isense.com/> [2000, April 27].
- Irani, M., Rousse, B., & Peleg, S. (1997). Recovery of ego-motion using region alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3), 6 pp.
- Jacobs, M. C., Livingston, M. A., & State, A. (1997). Managing latency in complex augmented reality systems, *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (pp. 49-54). Providence, RI.
- Jacobs, O. L. R. (1993). *Introduction to Control Theory* (Second ed.): Oxford University Press.
- Janin, A., Zikan, K., Mizell, D., Banner, M., & Sowizral, H. (1994). A videometric head-tracker for augmented reality applications, *Telemanipulator and Telepresence Technologies* (Vol. 2351, pp. 308-315). Bellingham, WA.
- Jebara, T., Eyster, C., Weaver, J., Starner, T., & Pentland, A. (1997). An optical tracker for augmented reality and wearable computers, *Proceedings of the First International Symposium on Wearable Computers* (pp. 146-150). Cambridge, MA.

- Julier, S., J., & Uhlmann, J., K. (1996). *A General Method for Approximating Nonlinear Transformations of Probability Distributions* (Technical Report). Oxford, UK: Robotics Research Group, Department of Engineering Science, University of Oxford.
- Julier, S., J., Uhlmann, J., K., & Durrant-Whyte, H., F. (1995). A New Approach for Filtering Nonlinear Systems, *1995 American Control Conference* (pp. 628-1632).
- Kadaba, M. P., & Stine, R. (2000). *Real-Time Movement Analysis Techniques and Concepts for the New Millennium in Sports Medicine*, [HTML]. Motion Analysis Corporation, Santa Rosa, CA USA. Available: <http://www.motionanalysis.com/applications/movement/rтанalysis.html> [2000, September 15].
- Kailath, T., Sayed, A., H., & Hassibi, B. (2000). *Linear Estimation*. Upper Saddle River, NJ USA: Prentice Hall.
- Kalawsky, R. S. (1993). *The Science of Virtual Reality and Virtual Environments* (First ed.): Addison-Wesley Publishing Company.
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME—Journal of Basic Engineering*, 82(Series D), 35-45.
- Kanade, T. Very fast 3-D sensing hardware (pp. 15 pp).
- Kaplan, E. D. E. (1996). *Understanding GPS Principles and Applications*: Artech House.
- Kim, D., Richardst, S., W., & Caudellt, T., P. (1997). An Optical Tracker for Augmented Reality and Wearable Computers, *1997 Virtual Reality Annual International Symposium (VRAIS '97)* (pp. 146-150). Albuquerque, NM: IEEE.
- Kite, D. H., & Magee, M. (1990). Determining the 3D position and orientation of a robot camera using 2D monocular vision. *Pattern Recognition*, 23(8), 819-831.
- Klinker, G. J., Ahlers, K. H., Breen, D. E., Chevalier, P.-Y., Crampton, C., Greer, D. S., Koller, D., Kramer, A., Rose, E., Tuceryan, M., & Whitaker, R. T. (1997). Confluence of computer vision and interactive graphics for augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 433-451.
- Kolasinski, E. M. (1995). *Simulator sickness in virtual environments* (Technical report 1027). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.
- Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R., & Tuceryan, M. Real-time vision-based camera tracking for augmented reality applications (pp. 8 pp).
- Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R., & Tuceryan, M. (1997a). Automated camera calibration and 3D egomotion estimation for augmented reality applications, *Proceedings of the CAIP '97, 7th International Conference Computer Analysis of Images and Patterns* (pp. 199-206).
- Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R., & Tuceryan, M. (1997b). Real-time vision-based camera tracking for augmented reality applications, *Proceedings of the ACM Symposium on Virtual Reality, Software and Technology (VRST-97)*.
- Krause, L. O. (1987). A direct solution to GPS-type navigation equations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(2), 225-232.
- Krauss, T. (1997). Beware of shortcomings when applying classical spectral-analysis techniques. *Personal Engineering*, 36-42.

- Krouglcof, N., McKinnon, G. M., & Svoboda, J. (1987). Optical position and orientation measurement techniques. United States: CAE Electronics, Ltd., Montreal, Canada.
- Krouglcof, N., Svoboda, J. V., & McKinnon, G. M. (1986). Noncontact position and orientation measurement techniques for real-time systems, *Proceedings of ASME International Computers in Engineering Conference* (pp. 177-183).
- Kubach, U., Leonhardi, A., Rothermel, K., & Schwehm, M. (1999). *Analysis of distribution schemes for the management of location information*. Stuttgart: Institut fur Parallelle und Verteilte, Hochstleistungrechner (IPVR), Universitat Stuttgart.
- Kuipers, J. B. (1980). SPASYN - An electromagnetic relative position and orientation tracking system. *IEEE Transactions on Instrumentation and Measurement, IM-29*(4), 462-466.
- Kuipers, J. B. (1998). *Quaternions and Rotation Sequences*. Princeton: Princeton University Press.
- Kumar, R., & Hanson, A. R. (1994). Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Understanding, 60*(3), 313-342.
- Kutulakos, K. N., & Vallino, J. R. (1998). Calibration-free augmented reality. *IEEE Transactions on Visualization and Computer Graphics, 4*(1), 1-20.
- Laughlin, D. R., Ardaman, A. A., & Sebesta, H. R. (1992). Inertial angular rate sensors: theory and applications. *Sensors*(October), 20-24.
- Lee, J. W., & Neuman, U. (2000). Motion Estimation with Incomplete Information using Omnidirectional Vision, *ICIP2000*.
- Lee, J. W., You, S., & Neuman, U. (2000). Large Motion Estimation for Omnidirectional Vision, *IEEE Workshop on Omnidirectional Vision 2000*.
- Leonhardi, A., & Kubach, U. (1999). An Architecture for a Distributed Universal Location Service, *Proceedings of the European Wireless '99 Conference* (pp. 351-355). Munich, Germany: ITG Fachbericht, VDE Verlag.
- Lewis, F. L. (1986). *Optimal Estimation with an Introductory to Stochastic Control Theory*. John Wiley & Sons, Inc.
- Lindgren, A. G., & Gong, K. F. (1978). Position and velocity estimation via bearing observations. *IEEE Transactions on Aerospace and Electronic Systems, AES-14*(4), 564-577.
- Link, B. (1993). Field-qualified silicon accelerometers: from 1 milli g to 200,000 g. *Sensors, 28*-33.
- List, U. H. (1983). *Nonlinear prediction of head movements for helmet-mounted displays* (Final technical paper): Operations Training Division, Williams Air Force Base, Arizona.
- Livingston, M. A., & State, A. (1997). Magnetic tracker calibration for improved augmented reality registration. *Presence: Teleoperators and Virtual Environments, 6*(5), 532-546.
- Lucas, B. D., & Kanade, T. (1985). Optical navigation by the method of differences. In A. K. Joshi (Ed.), *Proceedings of the 9th International Joint Conference on Artificial Intelligence* (pp. 981-984): Morgan Kaufmann.

- MAC. (2000). *HiRes 3D Motion Capture System*, [html]. Motion Analysis Corporation.
Available: <http://www.motionanalysis.com/applications/movement/gait/3d.html>
[2000, September 15].
- Macellari, V. (1983). CoSTEL: a computer peripheral remote sensing device for 3-dimensional monitoring of human motion. *Medical & Biological Engineering & Computing*, 21, 311-318.
- Madhavan, R., Dissanayake, M., & Durrant-Whyte, H., F. (1998). Map-building and map-based localization in an underground-mine by statistical pattern matching. In S. V. A.K. Jain, B.C. Lovell (Ed.), *International Conference on Pattern Recognition (ICPR)* (Vol. 2, pp. 1744-1746). Brisbane, Australia: IEEE Computer Society Press Piscataway, NJ USA.
- Mahmoud, R., Loffeld, O., & Hartmann, K. (1994). Multisensor Data Fusion for Automated Guided Vehicles, *Proceedings of SPIE - The International Society for Optical Engineering* (Vol. 2247, pp. 85-89).
- Mann, R. W., Rowell, D., Dalrymple, G., Conati, F., Tetewsky, A., Ottenheimer, D., & Antonsson, E. (1983). Precise, rapid, automatic 3-D position and orientation tracking of multiple moving bodies. In H. Matsui & K. Kobayashi (Eds.), *Proceedings of the VIII international congress of biomechanics* (pp. 1104-1112). Champaign, IL: Human Kinetics.
- Mark, W. R., McMillan, L., & Bishop, G. (1997). Post-rendering 3D warping, *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (pp. 7-16). Providence, RI.
- Matthies, L., & Shafer, S. A. (1987). Error Modeling in Stereo Navigation. *IEEE Journal of Robotics and Automation*, RA-3(3), 239-248.
- Maybeck, P. S. (1979). *Stochastic models, estimation, and control* (Vol. 141).
- Mazuryk, T., & Gervautz, M. (1995). Two-Step Prediction and Image Deflection for Exact Head Tracking in Virtual Environments, *Proceedings of EUROGRAPHICS 95* (EUROGRAPHICS 95 ed., Vol. 14 (3), pp. 30-41).
- McFarland, R. E. (1986). *CGI delay compensation* (NASA Technical Memorandum NASA TM 86703): NASA Scientific and Technical Information Branch.
- McFarland, R. E. (1988). *Transport delay compensation for computer-generated imagery systems* (NASA Technical Memorandum NASA TM 100084). Moffett Field, CA: NASA Ames Research Center.
- Meditch, S. (1969). *Stochastic Optimal Linear Estimation and Control*. New York: McGraw-Hill.
- Meyer, K., Applewhite, H., & Biocca, F. (1991). A survey of position trackers (pp. 61).
- Meyer, K., Applewhite, H. L., & Biocca, F. A. (1992). A Survey of Position Trackers. *Presence, a publication of the Center for Research in Journalism and Mass Communication*, 1(2), 173-200.
- Mine, M. R. (1993). *Characterization of end-to-end delays in head-mounted display systems* (Technical report TR93-001). Chapel Hill, NC: Department of Computer Science, University of North Carolina at Chapel Hill.

- Molineros, J., Raghavan, V., & Sharma, R. (1998). Computer vision based augmented reality for guiding and evaluating assembly sequences (poster), *Proceedings of IEEE VRAIS '98* (pp. 214). Atlanta, GA.
- Mulder, A. (1994a). *Human Movement Tracking Technology* (Technical Report TR 94-1): School of Kinesiology, Simon Fraser University.
- Mulder, A. (1994b, May 8, 1998). *Human Movement Tracking Technology: Resources*, [HTML]. School of Kinesiology, Simon Fraser University. Available: <http://www.cs.sfu.ca/people/ResearchStaff/amulder/personal/vmi/HMTT.add.html> [2000, September 15].
- Mulder, A. (1998, May 8, 1998). *Human Movement Tracking Technology*, [HTML]. School of Kinesiology, Simon Fraser University. Available: <http://www.cs.sfu.ca/people/ResearchStaff/amulder/personal/vmi/HMTT.pub.html> [2000, September 15].
- Nardone, S. C., & Aidala, V. J. (1981). Observability criteria for bearings-only target motion analysis. *IEEE Transactions on Aerospace and Electronic Systems, AES-17*(2), 162-166.
- Nash, J. (1997, October 1997). Wiring the jet set. *Wired*, 5, 128-135.
- Nayar, S. K., Watanabe, M., & Noguchi, M. (1994). *Real-time focus range sensor* (paper CUCS-028-94). New York, NY: Department of Computer Science, Columbia University.
- NDI. (2001). *OPTOTRAK*, [HTML]. Northern Digital Inc. Available: <http://www.ndigital.com/optotrk.html> [2001, April].
- Nebot, E., M. (1999). Sensors Used for Autonomous Navigation. In S. G. Tzafestas (Ed.), *ADVANCES IN INTELLIGENT AUTONOMOUS SYSTEMS* (pp. 135-156): Kluwer Academic Publisher (Dordrecht / Boston / London).
- Nebot, E., M., Durrant-Whyte, H., F., & Scheding, S. (1988). Frequency domain modelling of aided GPS for vehicle navigation systems. *Robotics and Autonomous Systems*, 25(1), 73-82.
- Neilson, P. D. (1972). Speed of Response or Bandwidth of Voluntary System Controlling Elbow Position in Intact Man. *Medical and Biological Engineering*, 10, 450-459.
- Nettleton, E. W., Gibbens, P. W., & Durrant-Whyte, H., F. (2000). Closed form solutions to the multiple platform simultaneous localisation and map building (SLAM) problem, *AeroSense 2000*. Orlando, FL USA.
- Neumann, U., & Cho, Y. (1996). A self-tracking augmented reality system, *ACM International Symposium on Virtual Reality and Applications* (pp. 109-115). Hong Kong.
- Neumann, U., & Majoros, A. (1998). Cognitive, performance, and system issues for augmented reality applications in manufacturing and maintenance, *Proceedings of IEEE VRAIS '98* (pp. 4-11). Atlanta, GA.
- Neumann, U., & Park, J. (1998). Extendible object-centric tracking for augmented reality, *Proceedings of IEEE VRAIS '98* (pp. 148-155). Atlanta, GA.
- Neumann, U., & You, S. (1998). Integration of region tracking and optical flow for image estimation.

- Neumann, U., & You, S. (1999). Natural Feature Tracking for Augmented Reality. *IEEE Transactions on Multimedia*, 1(1), 53-64.
- Newman, J. (1999). An example of the extended Kalman filter (pp. 7 pp).
- Oceanographers, W. (2001). *Women Exploring the Oceans*, [HTML]. Available: <http://www.womenoceanographers.org> [2001, April 27].
- Ohshima, T., Satoh, K., Yamamoto, H., & Tamura, H. (1998). AR2 hockey: a case study of collaborative augmented reality, *Proceedings of IEEE VRAIS '98* (pp. 268-275). Atlanta, GA: IEEE.
- Pasman, W., van der Schaaf, A., Lagendijk, R., L., & Jansen, F., W. (1999). Accurate overlaying for mobile augmented reality. *Computers & Graphics*, 23, 875-881.
- Petridis, V. (1981). A method for bearings-only velocity and position estimation. *IEEE Transactions on Automatic Control*, AC-26(2), 488-493.
- Phillips, D. (2000, April). On the Right Track—A unique optical tracking system gives users greater freedom to explore virtual worlds. *Computer Graphics World*, 16-18.
- Polhemus. (2000). *Polhemus*, [HTML]. Polhemus. Available: <http://www.polhemus.com/home.htm> [2000, September 15].
- Raab, F. H., Blood, E. B., Steiner, T. O., & Jones, H. R. (1979). Magnetic Position and Orientation Tracking System. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15, 709-718.
- Rae-Dupree, J. (1997, July 26). Experts Look at Where Computing is Headed; Leaving the Desktop Behind. *San Jose Mercury News*, pp. 1.
- Rauch, H. E. (1963). Solutions to the Linear Smoothing Problem. *IEEE Transactions on Auto. Control*, AC-8, 371-372.
- Rauch, H. E., Tung, F., & Striebel, C. T. (1965). Maximum Likelihood Estimates of Linear Dynamic Systems. *AIAA Journal*, 3, 1445-1450.
- Rehg, J. M., & Kanade, T. (1993). *DigitEyes: vision-based human hand tracking* (CMU-CS-93-220). Pittsburgh, PA: School of Computer Science, Carnegie Mellon University.
- Rehg, J. M., & Kanade, T. (1994). Visual tracking of high dof articulated structures: An application to human hand tracking. In J.-O. Eklundh (Ed.), *Proceedings of European Conference on Computer Vision* (Vol. 2, pp. 35-46): Springer-Verlag.
- Rekimoto, J. (1997). NaviCam: a magnifying glass approach to augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 399-412.
- Rekimoto, J., & Nagao, K. (1995). The World through the Computer: Computer Augmented Interaction with Real World Environments, *1995 Symposium on User Interface Software and Technology (UIST 95)*. Pittsburgh, PA: Association of Computing Machinery.
- Reunert, M. K. (1993). Fiber-optic gyroscopes: principles and applications. *Sensors*(August), 37-38.
- Richards, J. (1998). The Measurement of Human Motion: A Comparison of Commercially Available Systems, *3D Conference of Human Movement*. University of Tennessee, Chattanooga, TN USA.

- Richards, J. (1999). The Measurement of Human Motion: A Comparison of Commercially Available Systems. *Human Movement Sciences*, 18.
- Riner, B., & Browder, B. (1992). Design guidelines for a carrier-based training system, *IMAGE VI Conference* (pp. 65-73). Scottsdale, Arizona.
- Roberts, K. S., & Ganapathy, S. (1987). *Stereo Triangulation Techniques* (Technical Memorandum 11352-861121-07TM). Holmdel, NJ: AT&T Bell Laboratories.
- Roberts, L. G. (1966). The Lincoln Wand, *Proceedings of the 1966 Fall Joint Computer Conference, AFIPS Conference Proceedings* (Vol. 29, pp. 223-227).
- Robinett, W., & Holloway, R. (1994). *The virtual display transformation for virtual reality*. Chapel Hill, NC, USA: University of North Carolina at Chapel Hill, Department of Computer Science.
- Rose, E. J., Bose, S. C., Kouba, J. T., & Sobek, D. A. (1985). *A cost/performance analysis of hybrid inertial/externally referenced positioning/orientation systems* (ETL-R-086). Fort Belvoir, VA: U.S. Army Engineer Topographic Laboratories.
- Salz, J., & Netravali, A. N. (1983). *Algorithms for estimation of 3-D motion* (Technical Memorandum TM 11345-831110-17, 1138-831110-01). Bell Laboratories.
- Sawhney, H. S. (1994). Simplifying motion and structure analysis using planar parallax and image warping, *International Conference on Pattern Recognition, 1994* (pp. 14 pp). Jerusalem.
- Sawhney, H. S., Ayer, S., & Gorkani, M. (1994). Model-based 2D & 3D dominant motion estimation for mosaicing and video representation (pp. 31 pp).
- Sawhney, H. S., Ayer, S., & Gorkani, M. (1995). Model-based 2D & 3D dominant motion estimation for mosaicing and video representation, *Proceedings of the IEEE International Conference on Computer Vision* (pp. 583-590). Cambridge, MA.
- Scheding, S., Dissanayake, M., Nebot, E., M., & Durrant-Whyte, H., F. (1999). An experiment in autonomous navigation of an underground mining vehicle. *IEEE Transactions on Robotics & Automation*, 15(1), 85-95.
- Schodl, A., Haro, A., & Essa, I. A. (1998). *Head tracking using a textured polygonal model* (Technical report GIT-GVU-98-24). Atlanta, GA: Georgia Institute of Technology, College of Computing, Graphics, Visualization and Usability Center.
- Schut, G. B. (1960). On exact linear equations for the computation of the rotational elements of absolute orientation. *Photogrammetria*, 16(1), 34-37.
- Selspot. (1987). *Selspot MULTILab system description* (1987-02-05).
- Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27, 379-423.
- Sharma, R., & Molineros, J. (1997). Computer vision-based augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(3), 292-317.
- Shaw, C., & Liang, J. (1992). An experiment to characterize head motion in VR and RR using MR, *Proceedings of the 1992 Western Computer Graphics Symposium* (pp. 99-101).

- Shoemake, K. (1985). Animating Rotation with Quaternion Curves, *Computer Graphics* (SIGGRAPH 85 Conference Proceedings ed., pp. 245-254). San Francisco: ACM Press.
- Smith, B. R. (1984). Digital head tracking and position prediction for helmet mounted visual display systems, *AIAA 22nd Aerospace Sciences Meeting*. New York, NY: AIAA.
- So, R. H. Y., & Griffin, M. J. (1992a). Compensating lags in head-coupled displays using head position prediction and image deflection. *Journal of Aircraft*, 29(6), 1064-1068.
- So, R. H. Y., & Griffin, M. J. (1992b). *Selspot Technical Specifications, Selcom Laser Measurements*.
- Soatto, S., Perona, P., Frezza, R., & Picci, G. (1996). Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3), 393-414.
- Sorensen, B. R., Donath, M., Yang, G.-B., & Starr, R. C. (1989). The Minnesota Scanner: a prototype sensor for three-dimentional tracking of moving body segments. *IEEE Transactions on Robotics and Automation*, 5(4), 499-509.
- Sorenson, H. W. (1970, July). Least-Squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7, 63-68.
- Sowizral, H., & Barnes, D. (1993). Tracking Position and Orientation in a Large Volume, *Proceedings of IEEE VRAIS 93* (pp. 132-139): IEEE Computer Society Press.
- SPIRIT. (2000). *SPIRIT Project: Making networks location-aware*. Available: <http://www.uk.research.att.com/spirit/> [2000, January 4].
- Spohrer, J. (1999a). 3. Key subproblem: determining location, [HTML]. Learning Communities Group, ATG, (c)Apple Computer, Inc. Available: <http://worldboard.org/pub/spohrer/wbconcept/S03.html> [1999, December 24, 1999].
- Spohrer, J. (1999b). Information in Places. *IBM Systems Journal, Pervasive Computing*, 38(4).
- Spohrer, J. (1999c, June 16). *WorldBoard: What Comes After the WWW?*, [HTML]. Learning Communities Group, ATG, (c)Apple Computer, Inc. Available: <http://worldboard.org/pub/spohrer/wbconcept/default.html> [1999, December 24, 1999].
- State, A., Hirota, G., Chen, D. T., Garrett, B., & Livingston, M. A. (1996). Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. In H. Rushmeier (Ed.), *SIGGRAPH 96 Conference Proceedings* (pp. 429-438): Addison Wesley.
- Strickland, D., Patel, A., Stovall, C., Palmer, J., & McAllister, D. Self tracking of human motion for virtual reality systems (pp. 10 pp).
- Strickland, D., Patel, A., Stovall, C., Palmer, J., & McAllister, D. (1994). Self tracking of human motion for virtual reality systems, *Proceedings of the Stereoscopic Displays and Virtual Reality Systems* (Vol. 2177, pp. 278-287). Bellingham, WA: SPIE.
- Sutherland, I. E. (1968). A head-mounted three dimensional display, *Proceedings of the 1968 Fall Joint Computer Conference, AFIPS Conference Proceedings* (Vol. 33, part 1, pp. 757-764). Washington, D.C.: Thompson Books.

- Sutherland, I. E. (1974). Three-dimensional data input by tablet, *Proceedings of the IEEE* (Vol. 62 (4), pp. 453-461).
- Tarabanis, K. A., Allen, P. K., & Tsai, R. Y. (1995). A survey of sensor planning in computer vision. *IEEE Trans. Robotics and Automation*, 11, 86-104.
- Thompson, E. H. (1959). An exact linear solution of the problem of absolute orientation. *Photogrammetria*, 15(4), 163-179.
- Thrun, S., Fox, D., & Burgard, W. (1998). A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Machine Learning*, 31, 29-53.
- Trucco, E., & Verri, A. (1998). *Introductory techniques for 3-D computer vision* (1st ed. Vol. 1). Upper Saddle River, New Jersey: Prentice Hall.
- UNC Tracker Project. (2000, July 10). *Wide-Area Tracking; Navigation Technology for Head-Mounted Displays*, [HTML]. Available: <http://www.cs.unc.edu/~tracker> [2000, July 18].
- Van Pabst, J. V. L., & Krekel, P. F. C. (1993). Multi Sensor Data Fusion of Points, Line Segments and Surface Segments in 3D Space, *7th International Conference on Image Analysis and Processing* – (pp. 174-182). Capitolo, Monopoli, Italy: World Scientific, Singapore.
- Varona, J. G., Roca, F. X., & Villanueva, J. J. (2000). iTrack: Image-based probabilistic tracking of people, *Proceedings of the International Conference on Pattern Recognition (ICPR'00)* (Vol. 3, pp. 1122-1125). Barcelona, Spain.
- Verplaetse, C., James. (1996). Inertial proprioceptive devices: Self-motion-sensing toys and tools. *IBM Systems Journal*, 35(3 & 4).
- Verplaetse, C. J. (1997). *Inertial-Optical Motion-Estimating Camera for Electronic Cinematography*. Unpublished Masters of Science, Massachusetts Institute of Technology, Cambridge, MA, USA.
- Vicci, L. (2001). *Quaternions and Rotations in 3-Space: The Algebra and its Geometric Interpretation* (Technical Report TR01-014). Chapel Hill, NC: University of North Carolina at Chapel Hill.
- Vieuille, T., Clergue, E., & Façao, P. E. D. S. (1993). Computation of ego-motion and structure from visual and inertial sensors using the vertical cue, *IEEE Proceedings of the Third International Conference on Computer Vision* (pp. 591-598). Berlin, Germany.
- VRPN. (2001). *Virtual Reality Peripheral Network*, [HTML]. 3rdTech. Available: <http://www.cs.unc.edu/Research/vrpn/> [2001, April 25].
- Wallmark, J. T. (1957). A new semiconductor photocell using lateral photo-effect. *Proceedings IRE*, 45, 474-483.
- Wang, J.-f. *A real-time optical tracker using off-the-shelf components*. Unpublished Dissertation proposal, University of North Carolina at Chapel Hill, Chapel Hill, NC.
- Wang, J.-F. (1990). *A real-time optical 6D tracker for head-mounted display systems*. Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC USA.

- Wang, J.-F., Azuma, R. T., Bishop, G., Chi, V., Eyles, J., & Fuchs, H. (1990). Tracking a Head-Mounted Display in a Room-Sized Environment with Head-Mounted Cameras, *SPIE 1990 Technical Symposium on Optical Engineering and Photonics in Aerospace Sensing* (Vol. 1290, pp. 47-57). Orlando, FL: SPIE.
- Wang, J.-f., Chi, V., & Fuchs, H. (1990). A Real-time Optical 3D Tracker for Head-mounted Display Systems, *Symposium on Interactive 3D Graphics* (I3D 90 Symposium Proceedings ed., Vol. 24 (2), pp. 205-215). Snowbird, UT: ACM Press, Addison Wesley.
- Ward, M., Azuma, R. T., Bennett, R., Gottschalk, S., & Fuchs, H. (1992). A Demonstrated Optical Tracker With Scalable Work Area for Head-Mounted Display Systems, *Symposium on Interactive 3D Graphics* (I3D 99 Conference Proceedings ed., pp. 43-52). Cambridge, MA USA: ACM Press, Addison-Wesley.
- Ware, C., & Balakrishnan, R. (1994). Target acquisition in fish tank VR: the effects of lag and frame rate, *Proceedings of Graphics Interface '94* (pp. 1-7).
- Warnekar, C. S., & Schalkoff, R. J. (1982). A predictor-corrector approach to tracking 3-D objects using perspective-projected images, *Proceedings of the IEEE Southeastcon '82* (pp. 371-384).
- Watanabe, K., Kobayashi, K., & Munekata, F. (1994). Multiple sensor fusion for navigation systems, *1994 Vehicle Navigation & Information Systems Conference Proceedings* (pp. 575-578): IEEE.
- Weber, H. (1997, April 9). *Predictive Head Tracking Using a Body-centric Coordinate System*. University of North Carolina at Chapel Hill. Available: <http://www.cs.unc.edu/~weberh/research/predtrac.html> and [Weber1997_pred_tracking.pdf](http://www.cs.unc.edu/~weberh/research/Weber1997_pred_tracking.pdf) [2001, January 21].
- Wefald, K. M., & McClary, C. (1984). Autocalibration of a laser gyro strapdown inertial reference/navigation system. *IEEE*, 66-74.
- Welch, G. (1995). *Hybrid Self-Tracker: An Inertial/Optical Hybrid Three-Dimensional Tracking System* (TR95-048). Chapel Hill, NC, USA: University of North Carolina at Chapel Hill, Department of Computer Science.
- Welch, G. (1996). *SCAAT: Incremental Tracking with Incomplete Information*. Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.
- Welch, G., & Bishop, G. (1995). *An Introduction to the Kalman Filter* (TR95-041). Chapel Hill, NC, USA: University of North Carolina at Chapel Hill, Department of Computer Science.
- Welch, G., & Bishop, G. (1997). SCAAT: Incremental Tracking with Incomplete Information. In T. Whitted (Ed.), *Computer Graphics* (SIGGRAPH 97 Conference Proceedings ed., pp. 333-344). Los Angeles, CA, USA (August 3 - 8): ACM Press, Addison-Wesley.
- Welch, G., & Bishop, G. (2001, December 8, 2000). *The Kalman Filter*, [HTML]. University fo North Carolina at Chapel Hill. Available: <http://www.cs.unc.edu/~welch/kalman/> [2001, January 1].

- Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., & Colucci, D. n. (1999). The HiBall Tracker: High-Performance Wide-Area Tracking for Virtual and Augmented Environments, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (pp. 1-11). University College London, London, United Kingdom (December 20 - 23): ACM SIGGRAPH, Addison-Wesley.
- Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., & Colucci, D. n. (2001). High-Performance Wide-Area Optical Tracking—The HiBall Tracking System. *Presence: Teleoperators and Virtual Environments*, 10(1).
- White, P. R., & Garrett, R. E. (1976). A generalized interactive three dimensional input system, *CAD 76, Second International Conference and Exhibition on Computers in Engineering and Building Design* (pp. 275-282): IPC Science and Technology Press.
- Wildes, R. P. (1991). Direct recovery of three-dimensional scene geometry from binocular stereo disparity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 761-774.
- Williams, S. B., Newman, P., Dissanayake, M., & Durrant-Whyte, H., F. (2000). Autonomous underwater simultaneous localisation and map building, *2000 IEEE International Conference on Robotics and Automation*. San Francisco, CA USA: Institute of Electrical and Electronic Engineers, Inc.
- Wloka, M. M. (1995). Lag in Multiprocessor Virtual Reality. *PRESENCE: Teleoperators and Virtual Environments*, 4(1), 50-63.
- Woltring, H. J. (1974). New Possibilities for Human Motion Studies by Real-Time Light Spot Position Measurement. *Biotelemetry*, 1, 132-146.
- Woltring, H. J. (1976). Calibration and measurement in 3-dimensional monitoring of human motion by optoelectronic means. II. Experimental results and discussion. *Biotelemetry*, 3, 65-97.
- Wood, D. N., Azuma, D. I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D. H., & Stuetzle, W. (2000). Surface Light Fields for 3D Photography. In K. Akeley (Ed.), *Proceedings of SIGGRAPH 2000* (Vol. <http://www.cs.washington.edu/homes/daniel/siggraph2000-slf.pdf>, pp. 287-296): ACM Press / ACM SIGGRAPH / Addison Wesley Longman.
- You, S., Neumann, U., & Azuma, R. T. (1999a, March 13-17). *Hybrid Inertial and Vision Tracking for Augmented Reality Registration*. Paper presented at the IEEE Virtual Reality, Houston, TX USA.
- You, S., Neumann, U., & Azuma, R. T. (1999b). Hybrid Inertial and Vision Tracking for Augmented Reality Registration, *IEEE Virtual Reality* (pp. 260-267). Houston, TX USA.
- You, S., Neumann, U., & Azuma, R. T. (1999c, Nov/Dec). Orientation Tracking for Outdoor Augmented Reality Registration. *IEEE Computer Graphics and Applications*, 19, 36-42.
- Youngblut, C., Johnson, R. E., Nash, S. H., Wienclaw, R. A., & Will, C. A. (1996). Full body motion interfaces, *Review of Virtual Environment Interface Technology* (pp. 181-208). Alexandria, VA: Institute for Defense Analyses.

Yun, W., & Howe, R. T. (1992). Recent developments in silicon microaccelerometers. *Sensors*(October), 31-41.

Zikan, K., Curtis, W., D., Sowrizal, H., A., & Janin, A., L. (1994). A note on dynamics of human head motions and on predictive filtering of head-set orientations, *Telemanipulator and Telepresence Technologies*.

C. Related Papers

This appendix includes a sample of some relevant papers that we have permission to reproduce. See also the “Tracking Bibliography” on page 95, and the electronic bibliography version at

<http://www.cs.unc.edu/~tracker/ref/s2001/tracker/>
